IT-Projektverträge rechtssicher gestalten

Ein Überblick über die wichtigsten Regelungspunkte für IT-Projektverträge

von

Dr. Frank A. Koch

Stand: 2011

Rechtsanwalt Dr. Frank A. Koch
Maximilianstr. 54
80538 München
Tel: 089/ 221 330 und 089/ 112 339
E-Mail: koch@anwaltskanzlei-koch.de
Website:www.anwaltskanzlei-koch.info
Blog: itrecht.blogg.de

Einleitung

Die Mehrzahl der IT-Projekte kann nicht zum geplanten Zeitpunkt, im geplanten Budget und mit der spezifizierten Qualität abgeschlossen werden. Dies führt zu vermeidbaren und teilweise erheblichen Zusatzkosten und kann sogar das wirtschaftliche Schicksal des Unternehmens gefährden, das von einer bestimmten Anwendung und/oder einem bestimmten Anbieter abhängig ist.

Mit auf die jeweiligen IT-Projekte passgenau zugeschnittenen und kontrollfähigen Projektverträgen können diese Risiken deutlich reduziert werden. Allerdings genügt es hierfür nicht, einfach Standardverträge aus Mustersammlungen zu übernehmen. Die spezifischen Projektrisiken müssen im Projektvertrag angemessen abgebildet werden. Nur auf diese Weise kann die dauerhafte Nutzbarkeit der Applikation und ausreichender Schutz des betrieblichen IT-Systeme vor Angriffen aus dem Netz erreicht werden.

Die Geschäftsleitung muss ein unmittelbares Interesse an der richtigen Gestaltung der Projektverträge und der Projektsteuerung haben. Versäumnisse können nämlich zu einer persönlichen Haftung der Mitglieder der Geschäftsleitung gegenüber dem Unternehmen führen (Compliance). Das gilt bereits dann, wenn und soweit das erforderliche Überwachungssystem zur Risikofrüherkennung nicht oder nicht ausreichend eingerichtet und angewendet wurde. Zu diesen Risiken gehören auch grundsätzlich vorhersehbare Probleme, die aus vom Scheitern bedrohten IT-Projekten resultieren.

Das vorliegende Skript fasst die wichtigsten Regelungspunkte für die rechtssichere Gestaltung und Steuerung von IT-Projekten zusammen. Es gründet auf der Darstellung des Verfassers „IT-Projektrecht“ (Wiss. Springer Verlag 2007), in der die Regelungspunkte und Problemstellungen ausführlicher erläutert werden. Die vorliegende Zusammenfassung soll für die Vertragspraxis eine erste Orientierung geben.

Eingefügt wurde neben Aktualisierungen ein ausführlicher Abschnitt zur agilen Programmierung.

Inhalt

I. Gestaltung von IT-Projektverträgen 4

1. Anforderungsanalyse – Requirement Management 4
2. Abschluss des Projektvertrages 17
3. Dokumentation der Anbieterleistung 18
4. Spezifische Probleme der Software-Erstellung 19
5. Change Management 22
6. Mitwirkung des Auftraggebers 23
7. Projektabnahme 25
8. Rechte an Programmformaten 28
9. Service Level Agreements 29
10. Hardware-Wartung und Software-Pflege in IT-Projekten 32
11. Regelungen zur Projektbeendigung 36
12. Vertragsanpassungen 36
13. Sanierung von IT-Projekten 40

II. Leistungsstörungen im Projekt 44
1. Kundenrechte bei Anbieterverzug 44
2. Kundenrechte aus Mängeln der Anbieterleistung 45

III Sonderfälle Beispiele typischer IT-Projekte 50
1. Einführung von Enterprise Resource Planning Software 50
2. Outsourcing 52
3. Agile Software-Entwicklung 64

IV. Compliance-Haftung der Geschäftsleitung aus unzureichender vertraglicher Absicherung und Kontrolle von IT-Projekten 76

V. ITIL-Best Practices und ISO-Normen als Prüfmaßstab 78

I. Gestaltung von IT-Projektverträgen

Das Schicksal von IT-Projektverträgen entscheidet sich weitgehend bereits bei den Vertragsverhandlungen. Am falschen Ende spart hier, wer nur allgemein formulierte Anforderungen vereinbart. Erschwert wird hier nämlich die notwendige Prüfung, ob der Anbieter die benötigte Leistung auch tatsächlich erbracht hat. Der Kunde läuft außerdem Gefahr, dass die von ihm nachträglich als erforderlich festgestellte Konkretisierungen einzelner Leistungspunkte als teure Sonderwünsche behandelt werden oder sogar unausführbar sind.

Der Projektverlauf sollte in Abschnitte (z.V. Modulerstellung) aufgeteilt werden („Milestones“), deren Erreichen getrennt kontrollfähig ist.

Aus dem Projektvertrag muss sich auch ergeben, welche Nutzungsrechte dem Kunden an der Software zustehen sollen. Hier werden in der (auch fach-)anwaltlichen Beratung nicht immer die urheberrechtlichen Besonderheiten aus der mittlerweile vorherrschenden objektorientierten Programmierung beachtet. Schöpferisches Gestalten kann hier oft weitgehend entfallen, wenn nur schematisch Festlegungen der Programmeigenschaften nach Kundenvorgaben erfolgen.

Zu regeln ist weiter, welche Rechte der Kunde bei Mängeln der Anbieterleistung hat und wie er diese Rehte durchsetzen kann, wann also etwa konkret mit einer Mängelbeseitigung spätestens zu rechnen ist.

Wird Software nur zeitlich begrenzt überlassen, ist zu regeln, ob etwa Programmkopien auf Datenträger zurückzugeben sind und ob Programmkopien in Rechnern gelöscht werden müssen bzw. welche Kontrollrechte der Anbieter insoweit hat.

1. Anforderungsanalyse – Requirements Management

Der Kunde muss, bevor er Anbietern Leistungsaufträge erteilt, zunächst unter-untersuchen, welche Probleme er eigentlich mit der betrieblichen IT lösen will. Dies klingt trivialer als es in der Praxis ist. So sollte der Kunde zunächst klären, ob bestimmte Geschäftsprozesse, die historisch gewachsen und unnötig komplex sind, vereinfacht bzw. standardisiert werden können. Dieses „Business Process Reengineering“ ist oft wesentlich kostengünstiger (und schneller) durchzuführen als ein individuelles Programmieren entlang vorfindlicher Anwendungsstrukturen. Notwendig ist also eine Anforderungsanalyse.

Üblicherweise teilt sich diese Anforderungsanalyse in eine Analyse des Ist-Zustands und der Soll-Anforderungen (Soll-Konzept) auf.

Ist-Analyse

In der Ist-Analyse sind die eingesetzten IT-Komponenten und bestehenden Abläufe mit den erkennbaren Schwachstellen (z.B. zu späte Rechnungsstellung, häufige Differenzen in der Buchhaltung, hohe Durchlaufzeiten in der Fertigung und lange Lieferzeiten im Vertrieb[1]) zu beschreiben. Weiter sollte eine Bewertung des Ist-Zustands erfolgen. In der **Beschreibung** sind u.a. folgende Fragen zu beantworten:[2]

- Welche **Geschäftsprozesse** (wie z.B. Ausführung eines Fertigungsauftrags, Abwicklung einer Kundenbestellung) werden im Unternehmen eingesetzt ? Müssen dieselben Kundendaten mehrfach eingegeben werden ? Mit wievielen derartiger Aufträge pro Zeiteinheit (Tag/Woche/Monat/Quartal) ist zu rechnen und wird in der näheren Zukunft bei Nachfragesteigerungen zu rechnen sein (um Reserven im Mengengerüst festzulegen). Für alle verwendeten Formulare und Belege sind die **Datenfelder** mit Bezeichnung von Inhalt und Länge, Sortierkriterien, Nummernsysteme (z.B. Identnummern), Aufbewahrungsfristen, etc. festzuhalten. Zu klären ist, welche Geschäftsprozesse mit (kostengünstigerer) Standardsoftware in welchem Umfang abgedeckt werden können.[3]
- Wer ist für diese Prozesse zuständig ?
- Aus welchen **Komponenten** besteht die IT-Infrastruktur ?
- Welche Clients und Server sind vorhanden ?
- Wie sind diese konfiguriert ?
- **Mengengerüst**: Wo fallen welche Daten an ? Wer erfasst und wer bearbeitet welche Daten ? Wer erhält welche Auswertungen ? Das Mengengerüst legt die benötigte Dimensionierung von Speichermedien und Datenleitungen (Datendurchsatz) fest. Erwartbare Zuwächse an Datenmengen sind zu berücksichtigen. Zum typischen Mengengerüst gehören: a) Stammdaten und Änderungsdaten hierzu (z.B. Kunden- oder Lieferantenanschriften, Artikel), b) Bestandsdaten (Debitoren-/Kreditoren-/Sachkonten, Lagerpositonen, Arbeitszeitkonten), c) Bewegungsdaten (z.B. Kundenaufträge, Bestellungen bei Lieferanten, Lagerentnahmen/zugänge, Kunden-/Lieferantenrechnungen, Zahlungseingänge/-ausgänge, Mahnungen).
- Vorhandene IT-**Komponenten** und Software-**Anwendungen**,[4] soweit diese weiter verwendet werden sollen. Prüfen: Werden solche Legacy-Systeme oder –Programme in der neuen Applikation unbedingt benötigt ?
- Welche **Schnittstellen** zu anderen internen Anwendungen und zu externen IT-Systemen (etwa der Finanzverwaltung) bestehen bzw. müssen eingerichtet werden ?
- Wie kann man die Prozesse mit Hilfe von Software weiter **optimieren** ?

---

1 *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 242.

2 *Stahlknecht*/*Hasenkamp*, a.a.O., 228; *Koch*, IT-Projektrecht, Rdn. 13.

3 *Streitz*, IT-Projekte retten, 29.

4 *Streitz*, IT-Projekte retten, 28.

- Sind die Software-**Lizenzen** im Unternehmen einheitlich und unternehmensweit geregelt ? Liegt für jeden Arbeitsplatz eine gültige Lizenz vor ? Können ungenutzte Lizenzen gekündigt werden ?
- Gibt es nicht benötigte Rechner/Lizenzen oder Installationen ?
- Ist das Patch-Management zentral organisiert ?

Der Kunde muss beachten, dass sich vorhandene, also ungelöst gebliebene Probleme auf der fachlichen Ebene nicht durch IT-Einsatz lösen lassen, sondern nur verlagert werden. Läuft beispielsweise ein Geschäftsprozess zu langsam ab, ist er nicht ausreichend transparent organisiert oder zu kostenträchtig, so hilft seine Abbildung in eine IT-Anwendung kein Stück weiter. Vor Beginn der Soll-Analyse und Erstellung von Lastem- und Pflichtenheft müssen erforderliche **Anpassungen** von **Geschäftsprozessen** durchgeführt werden. Erst dann ist die Basis konsolidiert, von der aus das IT-Projekt in Angriff genommen werden kann.

Soll-Analyse

In der – grundsätzlich vom Kunden durchzuführenden – Soll-Analyse ist darzulegen, welche Aufgaben zukünftig benötigt werden, auf welche Weise diese zu erledigen sind und welches Mengengerüst voraussichtlich benötigt werden wird. Hierbei ist vorab zu prüfen, ob und gegebenenfalls welche Geschäftsprozesse optimiert werden (oder u.U. entfallen) können. Aufgetretene Probleme (z.B. zu ungenaue Formulare), Fehlerquellen bei Ergebnissen oder Engpässe (etwa bei gleichzeitiger Bearbeitung verschiedener Aufträge durch mehrere Mitarbeiter, Stoßbelastungen) sind zu berücksichtigen, ebenso Effizienzmängel (etwa mehrfaches Erfassen von Daten auf den verschiedenen Stufen eines Geschäftsprozesses). Ein Teil dieser Optimierungen kann durch **Umorganisation** erfolgen (Business Reengineering), die vor Erstellung des Lasten-/Pflichtenhefts durchzuführen ist (damit dieses nicht von einer inzwischen überholten Basis ausgeht). Ein anderer Teil wird durch zu beschaffende neue Applikation zu leisten sein. Die Anforderungen an diese müssen komplett in der Leistungsvorgabe für den Anbieter erfasst sein.

Die Ergebnisse aus dieser Soll-Analyse sind in einem **Fachkonzept** darzulegen. In das Fachkonzept sind alle Geschäftsprozesse einzubeziehen. Die Abläufe müssen aus der fachlichen Sicht möglichst genau beschrieben werden. So muss es etwa möglich sein, einem Besteller eine vorhandene Kundennummer zuzuordnen, wenn der Besteller bereits früher Bestellungen getätigt hat. Weiter muss die Bearbeitung des Auftrags auch dann möglich sein, wenn nur ein Teil der bestellten Produkte vom Lager ausgeliefert werden kann und der Rest erst nach Anlieferung durch den Zulieferer. Soll der Kunde den Weg seiner Bestellung in ihrer Bearbeitung internetgestützt mitverfolgen können, muss dies zusätzlich implementiert und gegen unberechtigte Zugriffe (oder gar unberechtigte Änderungen) abgesichert werden.[5] Unterschiedliche Formate der Kundendaten sind zu vereinheitlichen sind, um sie durchgängig bearbeiten zu können. Hieraus werden dann primär die fachlichen Vorgaben und Anforderungen in einem **Lastenheft** (im Sinne der Praxis)/Pflichtenheft (im Sinne der Rechtsprechung)

[5] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 228; *Koch*, IT-Projektrecht, Rdn. 18.

detailliert aufgeschlüsselt.[6] Das Spezifizieren der zur Erfüllung der fachlichen Aufgaben erforderlichen IT (im IT-Pflichtenheft) liegt hingegen grundsätzlich in der Verantwortlichkeit des beauftragten Anbieters.[7]

Die Erstellung der Anforderungsanalyse ist grundsätzlich vom Kunden durchzuführen,[8] wenn nicht der Anbieter mit dieser Analyse (etwa als Studie) ausdrücklich gesondert beauftragt wird. Auch ohne solchen Auftrag muss der Anbieter aber auf für ihn erkennbare Unklarheiten hinweisen und bei der Formulierung der Bedürfnisse mitwirken,[9] jedenfalls dann, wenn der Kunde zu erkennen gibt, dass er dieser Unterstützung bedarf. Der Anbieter muss mangels abweichender Vereinbarung aber nicht von sich aus ohne gesonderte Vereinbarung die fachlichen Anforderungen des Kunden komplett untersuchen.

Requirements Management

Die Anforderungen können im Verlauf längerfristiger IT-Projekte an neue Gegebenheiten anzupassen sein. Diese Anlassung bedarf genauer Kordinierung, damit nicht z.B. bereits erbrachte Leistungsteile ihre Verwendbarkeit verlieren. Notwendig ist hier ein geregeltes „Requirements Management" (RM)[10]. Ziel muss es sein, dass auch nach Festlegen und Durchführen aller Ergänzungen und Änderungen für beide Seiten erkennbar bleibt, welche Leistungen geschuldet und welche erbracht sind. Hierzu müssen die zugrundeliegenden Anforderungs- und Lösungsbeschreibungen laufend fortgeschrieben werden. Im RM werden die kundenseitigen Anforderungen im Zusammenwirken beider Vertragsparteien definiert, spezifiziert und verifiziert, analysiert, vereinbart, einem Projekt zugewiesen und in diesem mit allen Änderungen verfolgt.[11]

Das RM sollte zwischen den Vertragsparteien als Form der Leistungserbringung vereinbart werden. Das Pflichtenprogramm des Anbieters weitet sich hierdurch aus. Er muss etwa sicherstellen, dass die fachlichen **Anforderungen** des Kunden (jedenfalls aus Anbietersicht) **vollständig, korrekt, konsistent, testbar, verständlich, notwendig, eindeutig, umsetzbar** und in einer einheitlichen Basis zusammengefasst sind[12] und auch durch alle Änderungen hindurch bleiben. Keinesfalls darf der Anbieter Kundenanforderungen einfach identisch übernehmen, da für den Anbieter das Risiko zu groß ist, eine Lösung in die falsche Richtung zu implementieren. Für die Gesamtheit der

---

[6] *Streitz*, IT-Projekte retten, 23. Nach VDI 2519 beinhaltet das Lastenheft die quantifizierbare und prüfbare, vom Auftraggeber als Ausschreibungs- oder Vertragsgrundlage zu erstellende Beschreibung aller Anforderungen aus Anwendersicht einschließlich aller Randbedingungen, also das „Was und Wofür", hingegen das Pflichtenheft die Beschreibung der Realisierung des Lastenhefts (also das „Wie").

[7] So etwa das OLG Köln CR 1994, 212.

[8] OLG Köln, Urt.v.29.7.2005 – 19 U 4/05, JurPC Web-Dok. 16/2006; OLG Köln NJW-RR 1995, 51f; 1993, 1529f.

[9] OLG Köln, Urt.v.29.7.2005, a.a.O.

[10] Ausführlich s. *Koch*, Requirements Management, IT-Rechtsberater 7/2009, 160.

[11] *Ebert*, Systematisches Requirements Management, 17.

[12] *Ebert*, Systematisches Requirements Management, 39; *Tiemeyer* (Hrg.), Handbuch IT-Management, 110.

Anforderungen ist mit dem Kunden eine **Priorisierung** durchzuführen, bei der die erfolgsentscheidenden Anforderungen vorangestellt werden. Nach der Überprüfung sollen im RM die so geprüften und überarbeiteten Anforderungen selbst als verbindlich **vereinbart**[13] und im später auf dieser Basis zu erstellenden IT-Pflichtenheft dokumentiert werden.

Alle während des Projekts durchzuführenden Anforderungs**änderungen** sind ausdrücklich zu vereinbaren – wobei der Anbieter auch das von ihm erstellte Pflichtenheft zu aktualisieren und die Änderung in der Ausführung zu dokumentieren hat. Parallel sollte der Kunde das von ihm erstellte Lastenheft regelmäßig aktualisieren, da es für ihn die einzige Grundlage der Leistungsabnahme ist. Bei längerlaufenden Projekten können mehrere Anpassungsläufe erforderlich werden.[14]

Das Erstellen von Lasten- und IT-Pflichtenheften ist kein starr fixierter Ablauf, sondern erfolgt bei Anbieter und Kunden jeweils in einem komplexen Prozess. Diese Prozesse müssen aufeinander abgestimmt erfolgen, was **mehrere Anpassungsdurchläufe** erforderlich machen kann.

Ziel des RM ist, „die Anforderungen an die im Projekt erstellten Produkte und Produktkomponenten zu managen und Inkonsistenzen zwischen diesen Anforderungen und den Projektplänen sowie den Arbeitsergebnissen zu identifizieren".[15] Das RM gründet sich wesentlich auf das Capability Maturity Model Integration (**CMMI**),[16] das seinerseits aus dem Capability Maturity Model (CMM) des Software Engineering Instituts (SEI) entwickelt wurde. Grundlage des CMM sind die Normen ISO 15504 für Assessment und Modelle der Prozessverbesserung, ISO 12207 für Lebenszyklen[17], ISO 15288 für Schnittstellen im Lebenszyklus und SPICE[18]. Das CMMI weist fünf **Reifegrade** auf:

„*Initial*" bzw. ad hoc (der Projekterfolg ist von Einzelinitiativen abhängig),
„*Gemanagt*" (Anforderungsmanagement, Projektmanagement wird für einzelne Projekte durchgeführt),
„*Definiert*" (einheitliche Prozesse für die gesamte Organisation, spezifizierte Anforderungsentwicklung),
„*Quantitativ gemanagt*" (durch statistische und quantitative Techniken) und
„*Optimierend*"[19].

CMMI ist stärker als ISO 9001 prozessorientiert (erleichtert also Kontrollen). Spezifisch auf das RM bezogen sind der IEEE-Standard 1233 für die Entwicklung und Spezifizierung von Anforderungen von Systemen, während der IEEE-Standard 830

---

13 *Ebert*, Systematisches Requirements Management, 18.

14 *Koch*, Requirements Management, IT-Rechtsberater 7/2009, 160, 161.

15 *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl. 2008, 568.

16 Ausführlich s. CMMI-Website www.sei.cmu.edu/cmmi.

17 *Ebert*, Systematisches Requirements Management, 32 m.w.N.

18 „SPICE" steht für Software Process Improvement and Capability Determination und soll der Bewertung der Software-Entwicklung dienen (*Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl. 2008, 5829.

19 *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl. 2008, 571.

spezifisch auf Software zugeschnitten ist.[20] Der Kunde sollte bei der Auftragsausschreibung einen möglichst hohen Reifegrad der Organisation des zu beauftragenden Anbieters zugrundelegen, denn je höher der Reifegrad ist, desto geringere Schwankungsbreite weisen die erzielten Ergebnisse im Verhältnis zu den Soll-Ergebnissen auf.[21] Anzuraten ist eine entsprechende **fallspezifische Kosten-Nutzen-Kalkulation**. Das **Requirements Management** ist erst **ab Reifegrad 3** organisiert (und verlässlich erwartbar), aber Voraussetzung für die Verfolgbarkeit von Anforderungen.[22] RM ist grundsätzlich auf das jeweilige gesamte IT-System (mit dessen Umgebung, etwa im Netz) ausgerichtet und nicht auf Software beschränkt.[23]

Im RM werden **Anforderungen** und **Lösungen** unterschieden. Alle Anforderungen, d.h. (Geschäfts-)Prozessanforderungen und funktionale wie nichtfunktionale Produktanforderungen müssen definiert, spezifiziert und verifiziert, analysiert, vereinbart und einem Projekt zugewiesen, im Projekt verfolgt und als Änderungen vereinbart werden.[24] Die Anforderungen müssen möglichst vollständig erfasst und eindeutig sowie konsistent formuliert werden, soll ihre Erfüllung überprüfbar sein.[25] Zu erfassen sind funktionale und nichtfunktionale Anforderungen. **Funktionale Anforderungen** beziehen sich auf die einzelnen Funktionen der Geschäftsprozesse oder sonstiger Anwendungen. Eine Schwachstelle im Projekt stellen erfahrungsgemäß **nichtfunktionale Anforderungen** dar. Sie werden mit teilweise eher vagen und jedenfalls nicht aus sich präzise in der Erfüllung überprüfbaren Eigenschaften wie „Benutzbarkeit“, „Verständlichkeit“, „Performanz“, „Qualität“, „Sicherheit“, „Wartbarkeit“, „Portierbarkeit“, oder „Zuverlässigkeit“ und beschrieben.[26] ISO/IEC 9126 führt u.a. folgende nichtfunktionale Eigenschaften an:[27]

- Interoperabilität (Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken),
- Ordnungsmäßigkeit (Erfüllung anwendungsspezifischer Normen, Vereinbarungen, gesetzlicher Bestimmungen etc.),
- (Daten-)Sicherheit,
- Zuverlässigkeit (Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen über einen festgelegten Zeitraum zu bewahren),
- Benutzbarkeit (Verständlichkeit, Erlernbarkeit und Bedienbarkeit),
- Effizienz (Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen, das Zeitverhalten und das Verbrauchsverhalten der benötigten Bedingungen),
- Änderbarkeit (Aufwand, der zur Durchführung von Änderungen wie Korrekturen, neue Funktionen notwendig ist),
- Portierbarkeit (Aufwand, die Software in eine andere Umgebung zu verlagern, dort zu installieren, anzupassen oder Teile auszutauschen).

---

20 *Ebert*, Systematisches Requirements Management, 36.

21 *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl. 2008, 573.

22 *Ebert*, Systematisches Requirements Management, 33.

23 *Koch*, Requirements Management, IT-Rechtsberater 7/2009, 160, 162.

24 *Ebert*, Systematisches Requirements Management, 17.

25 *Hindel*/*Hörmann*/*Müller*/*Schmied*, Basiswissen Software-Projektmanagement, 41.

26 *Ebert*, Systematisches Requirements Management, 98.

27 Nach: *Ebert*, Systematisches Requirements Management, 99.

Nichtfunktionale Anforderungen müssen operationalisiert und damit **messbar** und binär (als „nichterfüllt“ oder „erfüllt“) entscheidbar **gemacht** werden, damit ihre Einhaltung kontrollierbar [28] und darlegbar ist bzw. ein Mangel substantiiert behauptet werden kann.

Aus den Anforderungen erarbeitet der Anbieter eine **Lösung**. Die **Anforderung beschreibt** den zu **erreichenden Nutzen („**Problemraum**“), die Lösung** hingegen**,** wie dieser Nutzen exakt zu implementieren ist (**„Lösungsraum“**).[29] Dies entspricht in etwa der bisherigen Unterscheidung zwischen kundenseits zu erstellendem Lastenheft und anbieterseits zu erstellendem IT-Pflichtenheft. Schließlich muss der Anbieter die erarbeitete **Lösung** mit dem Kunden **abstimmen**, allein schon, weil die Beschreibung der Lösung zu neuen Fragen und weiteren Anforderungen führen kann. In „*Business Cases*“ bzw. „*Use Cases*“ (Benutzerszenarien) sollten alle möglichen Interaktionen mit dem System über äußere Schnittstellen beschrieben werden.[30] Festzustellen ist, welche Abhängigkeiten zwischen Anforderungen bestehen. Zu verfolgen ist außerdem, ob alle Anforderungen in Lösungsbeschreibungen und Testfällen abgebildet werden.

Während der Projektdurchführung sind auf jeder Stufe bestimmte Umsetzungsprozesse oder –merkmale zu verfolgen.
(1) Projektdefinition: Quellen für Änderungen;
(2) Systemanalyse/Entwurf: Analysestatus und Abdeckung;
(3) Implementierung/Verifikation: Status in Entwurf, Code und Verifikation; bei (1) bis (3) außerdem die Änderungshäufigkeit;
(4) Integration: Status, Qualität;
(5) Systemtest/Abnahme: Status, Abdeckung, Akzeptanz;
(6) Auslieferung/Wartung: Feldfehler, Änderungen.[31]

Diese Stufen sollten **im IT-Projektvertrag** ausdrücklich vereinbart werden.

Die Durchführung von Änderungen erfolgt als „Change Management“ grundsätzlich nach ITIL und ISO/IEC 20000.[32] Als „Change“ gilt jede Änderung an der IT-Infrastruktur eines Unternehmens.[33] Change Management soll die Änderungsanträge (Requests for Change, RfCs) steuern.[34] ISO 20000 enthält eine **integrierte und prozessorientierte Methodik für eine effektive Planung und Erstellung von IT-Services**. Die Norm folgt dem **PDC**A-Zyklus des **Plan** (Service Management planen), **Do** (Service Management implementieren), **Check** (Überwachen, Messen, [Über-]Prüfen) und **Act** (kontinuierlich verbessern).[35]

---

[28] *Ebert*, Systematisches Requirements Management, 103, 109.
[29] *Ebert*, Systematisches Requirements Management, 13.
[30] *Ebert*, Systematisches Requirements Management, 135, 121.
[31] *Ebert*, Systematisches Requirements Management, 75, 177.
[32] S. näher *Koch*, ITRB 2008, 61 und *Koch*, Requirements Management, IT-Rechtsberater 7/2009, 160, 162.
[33] *Victor*/*Günther*, Optimiertes IT-Management mit ITIL, 27.
[34] *Victor*/*Günther*, Optimiertes IT-Management mit ITIL, 57.
[35] ISO/IEC 20000-2 Nr. 4.1 – 4.4

Mit zunehmender Projektdauer wächst freilich die Anzahl der Änderungen und der Aufwand für ihr Management. Empfohlen wird deshalb, mittels einer Wirtschaftlichkeitsrechnung projektspezifisch einen Zeitpunkt festzulegen (und zu vereinbaren), ab dem die Anforderungen gewissermaßen eingefroren werden, da das Projekt durch die zunehmende Anzahl von Änderungen unwirtschaftlich wird („*Freeze*“).[36]

In größeren Projekten wird diese laufende Dokumentanpassung meist nur mittels **Versionierung** und als IT-gestütztes Document Management zu bewältigen sein. In auf Datenbankbasis zu implementierenden und laufend zu aktualisierenden *Project Reports* sollten alle Anpassungen mit Vereinbarungsdatum und Status „in Bearbeitung“, „zurückgewiesen“ (etwa bei nicht bestätigbaren Mängelbehauptungen), „zurückgestellt“, „behoben“ und „verifiziert“ mit automatisierten statistischen Auswertungen aufgelistet werden. Im Idealfall ist so für den Kunden zu jedem beliebigen Zeitpunkt der jeweils aktuelle Bearbeitungsstand abfragbar.[37] Das RM ist grundsätzlich vom Anbieter zu steuern. Dies stellt eine werkvertragliche Leistung des Anbieters dar, die vorsorglich gesondert **vereinbart** werden sollte.

Anhand klar definierter Kriterien muss für beide Seiten schnell feststellbar und entscheidbar sein, ob eine Änderung noch zum **Leistungsumfang** gehört, einem zugelassenen Sonderwunsch zuzuordnen ist oder eine nur gegen Zusatzkosten und/oder Terminsverlängerung durchzuführende Auftragserweiterung darstellt. Voraussetzung hierfür ist, dass eine spezifische Konfigurationsbasis („*Baseline*“) definiert und abgenommen ist (hier: im IT-Pflichtenheft), die nur durch einen formalen Änderungsprozess geändert werden kann. [38]

Geklärt und schon bei Vertragsschluss vereinbart werden muss, welche Anwendungsfälle mit welchen Datentypen zu testen sind und welche Testdaten der Kunde bis zu welchem Datum vorzugeben hat. **Testfälle** und –**daten** müssen alle vereinbarten Anforderungen abdecken. Es kann erforderlich sein, zunächst ein komplettes Testsystem zu installieren, bevor das System in die Phase der produktiven Nutzung übergehen kann („*Go live*“). Die Testdokumentation ist, wenn sie vom Anbieter durchzuführen ist, dessen zu vergütende Leistung.[39]

Leistungsbeschreibung

Die Leistungsbeschreibung ist neben dem Projektvertrag das wichtigste Dokument im Projekt. Die Leistungsbeschreibung sollte grundsätzlich möglichst **alle** relevanten **Leistungsteile** erfassen und konkret regeln. Der Kunde muss Position für Position durchprüfen und kontrollieren können, ob der Anbieter die jeweilige Leistung erbracht hat. Auch der Auftragnehmer muss Interesse an einer solchen Leistungsbeschreibung haben, da sie nicht nur festlegt, was er zu leisten hat, sondern auch, was **nicht** mehr zum

---

[36] *Ebert*, Systematisches Requirements Management, 179, 188.
[37] *Koch*, Requirements Management, IT-Rechtsberater 7/2009, 160, 162.
[38] *Koch*, a.a.O., 163.
[39] *Koch*, a.a.O., 163.

Leistungsumfang gehört und deshalb nur als zusätzlich zu vergütende Leistung erbracht zu werden braucht. Ebenso sind die (spätesten) **Leistungstermine** mit Datum festzulegen. Die Feststellungen aus dem Soll-Status sind in die Leistungsbeschreibung aufzunehmen, soweit die Leistungen vom Anbieter zu erbringen sind. Vom Auftraggeber zu erbringende Mitwirkungsleistungen sind getrennt hiervon im Vertrag oder in einer Anlage zu diesem festzuhalten; der Auftragnehmer sollte hier im eigenen Interesse diese Mitwirkungshandlungen möglichst klar bezeichnen, wenn er später etwa deren Nichterbringung behaupten muss.

Nur in dem Umfang, in dem eine solche präzise Leistungsbeschreibung erfolgt, können die Vertragsparteien in einem Rechtsstreit eine vereinbarte Beschaffenheit behaupten[40] (§ 633 Abs. 2 S. 1 BGB) und unter Beweis stellen (der Kunde, um deren Nicht- oder Schlechterfüllung zu behaupten, der Anbieter demgegenüber, um die vertragsgemäße Erfüllung darzulegen), während Beweisrisiken bezüglich der vertraglichen vorausgesetzten Verwendung (§ 633 Abs. 2 S. 2 Nr. 1 BGB) und erst recht bezüglich der gewöhnlichen Verwendung (§ 633 Abs. 2 S. 2 Nr. 2 BGB) auftreten können.[41]

Lastenheft und IT-Pflichtenheft als Formen der Leistungsbeschreibung

Das Lastenheft enthält grundsätzlich die fachlichen Anforderungen und ist vom Kunden zu erstellen. Das IT-Pflichtenheft enthält die vom Anbieter erstellte DV-technische Lösung des fachlichen Anwendungsproblems. Zu beachten ist, dass die Rechtsprechung das Lastenheft als „Pflichtenheft“ bezeichnet,[42] während in der IT-Praxis die fachlichen Anforderungen im „Lastenheft“[43], die DV-technische Lösung aber im „Pflichtenheft“ erfasst werden.

Die **endgültige Fassung** des Lastenhefts sollte wie auch später die des IT-Pflichtenhefts (schon aus Beweisgründen) als solche bezeichnet und von beiden Seiten **unterzeichnet**[44] werden. Die Vertragsparteien müssen beachten, dass Lasten- und Pflichtenheft die Basis für die spätere Entwicklung bilden; deshalb die offizielle **Freigabe** einer endgültigen Version wesentlich.[45]

---

[40] *Schneider*/v.Westphalen, Software-Erstellungsverträge, C Rdn. 82 f.

[41] *Koch*, IT-Projektrecht, Rdn. 26.

[42] *Schneider*/v.Westphalen, Software-Erstellungsverträge, C Rdn. 18f.

[43] Das „Lastenheft“ beinhaltet nach DIN 69901 die Gesamtheit der Anforderungen des Auftraggebers an die Lieferungen und Leistungen eines Auftragnehmers, das „Pflichtenheft“ die ausführliche Beschreibung der Leistungen, die erforderlich sind oder gefordert werden, damit die Ziele des Projekts erreicht werden. Die DIN-Norm nimmt keine Zuordnung der Verantwortlichkeiten vor (*Müller-Hengstenberg*, Der Vertrag als Mittel des Risikomanagements, CR 2005, 385, 390).

[44] *Klotz*/*Dorn*, Vertragsmanagement in der Informationsverarbeitung, 85.

[45] *Hindel*/*Hörmann*/*Müller*/*Schmied*, Basiswissen Software-Projektmanagement, 41.

Der **Beschreibungsumfang** des Lasten-/Pflichtenhefts (im Sinne von fachlichem Feinkonzept) muss umfassen:[46]

- Ist-Zustand (Ausgangssituation)
- Soll-Zustand (Zielbeschreibung)
- Beschreibung der Schnittstellen z.B. zwischen Benutzer oder Nicht-EDV-Funktionseinheiten (z.B. elektronischen Steuerungen), jeweils mit Beschreibung von Bildschirmein- und –ausgaben, von Inhalten der Information und von Formaten,
- Listenausgaben,
- Verarbeitungsregeln,
- Prüfregeln,
- Mengengerüste,
- Sicherung der Daten,
- zeitlicher Rahmen,
- Ergonomieanforderungen,
- benötigte Erweiterungsmöglichkeiten.

Wird vom Auftraggeber das von ihm zu erbringende Lasten-/**Pflichtenheft nicht erstellt**, scheitert hieran (wie beim Unterbleiben der Spezifikation einzelner Funktionen) die Auftragsausführung durch den Auftragnehmer nicht. Der Auftragnehmer ist vielmehr gehalten, eine Leistung zu erbringen, die nach den allgemeinen Grundsätzen dem Stand der Technik bei mittlerem Ausführungsstandard entspricht.[47] Zur Ausfüllung bestehender Lücken ist auf die **gewöhnliche Verwendung** der Software zurückzugreifen,[48] die freilich bei (lückenhaft definierter) individueller Anpassung ebenfalls schwer feststellbar ist. Gesetzliche Vorgaben an die Anwendung muss die Software vereinbarungsunabhängig erfüllen.[49]

Checkliste zur Erstellung des fachlichen Lastenhefts:

Zielvorgaben[50]

- **Vollständigkeit**: Sind alle relevanten Punkte geklärt und notwendigen Informationen eingeholt ?
- **Korrektheit**: Sind die eingeholten Informationen von dritter Seite bestätigt und im Lastenheft zutreffend dokumentiert ?
- **Aktualität**: Sind die verwendeten Informationen, Dokumente und Technologien aktuell ?

---

[46] Teilweise nach: *Müller-Hengstenberg*, BVB-Computersoftware, 177; *Balzert*, Lehrbuch der Software-Technik. Software-Entwicklung, 62.

[47] BGH, Urt.v.24.9.1991 – X ZR 85/90, CR 1992, 543 – „Vergessenes Pflichtenheft".

[48] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn. 285.

[49] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn. 286; *Koch*, Handbuch Software- und Datenbank-Recht, § 5 Rdn. 13.

[50] Nach *Grévent*/*Krömker*, Unklare Ziele gefährden Projekte, Computerwoche 2/2005, 30.

- **Eindeutigkeit**: Werden die Begriffe einheitlich und eindeutig verwendet ? Werden sie von den Vertragspartnern in derselben Weise verstanden ?
- **Detaillierungsgrad**: Sind die Anforderungen ausreichend detailliert ?

IT-Pflichtenheft

Auf der Basis des Lastenheft hat der Anbieter das technische Systemkonzept in der Form eines auf die IT bezogenen Pflichtenhefts zu erstellen.[51] Diese Erstellung ist entweder Teil des System- oder Software-Vertrags oder Gegenstand eines separaten Vertrages sein. Im zweiten Fall muss die Auftragsausführung zusätzlich vereinbart werden.

Das Pflichtenheft legt die **Sollbeschaffenheit** der vom Anbieter geschuldeten Leistung fest.[52] Das IT-bezogene Pflichtenheft ist integraler Teil der Erstellungs- oder Anpassungsleistung des Anbieters und kann meist nicht als getrennt abnahmefähige Teilleistung behandelt werden. Der Kunde kann beim bloßen Studium allein des IT-Pflichtenheft nämlich oft nicht entscheiden, ob es die zu erbringende Leistung richtig festlegt. Zu vereinbarende Teilzahlungen sollten deshalb nicht an der Abnahme des IT-Pflichtenhefts anknüpfen, sondern an dessen Übergabe.

Änderungen des Inhalts des Pflichtenhefts dürfen nur im klar geregelten Änderungs(Change Request-)verfahren zulässig sein.[53]

Grundsätzlich erfolgt die nähere Festlegung der Leistungskomponenten in der **Abfolge**: **Problemlösung** → **Software** → **Hardware**. Die erarbeitete Problemlösung legt also die auszuwählende bzw. zu erstellende Software fest und die Software ihrerseits die Hardware. Dies sollte auch im IT-Projektvertrag festgelegt werden.

Zu prüfen sind benötigte **Schnittstellen** zu vorhandenen Anwendungen,[54] Aufrüstbarkeit und Erweiterbarkeit[55]. Zu berücksichtigen ist auch der vorhandene **Datenbestand**, da sich sein Umfang auf den Ressourcenverbrauch und die anzuschaffende IT-Infrastruktur auswirken kann, [56] ebenso die Übernahme größerer **Altdatenbestände**.

Festzulegen sind bei objektorientierter Programmierung **Klassen**- und **Objektdefinitionen**. Das Pflichtenheft muss außerdem Betriebsbedingungen

---

[51] *Streitz*, IT-Projekte retten, 23. Das „Pflichtenheft“ umfasst nach DIN 69901 die vom Auftragnehmer erarbeiteten Realisierungsvorgaben aufgrund der Umsetzung des Lastenhefts.

[52] LG Trier, Urt.v.2.12.1992 – 5 O 1/92, CR 1995, 221 (für das fachbezogene Pflichtenheft).

[53] *Streitz*, IT-Projekte retten, 41.

[54] *Streitz*, IT-Projekte retten, 30.

[55] *Streitz*, IT-Projekte retten, 39.

[56] *Streitz*, IT-Projekte retten, 128.

spezifizieren, ebenso Qualitätsanforderungen, Benutzeroberflächen, technische Produkt- und Entwicklungsumgebungen.

Die Einbindung des Projekts in das **Qualitätssicherungssystem** des Anbieters ist im Projektvertrag zu regeln und im Pflichtenheft darzulegen. Während die Einrichtung des QS-Systems allgemein den Normen EN ISO 9000 folgt, müssen für die Software-Erstellung fachspezifische Qualitätsnormen für bestimmte Anwendungsbereiche gesondert bezeichnet und vereinbart werden. Die Bedeutung einer Qualitätsprüfung wird schnell deutlich, wenn man berücksichtigt, dass Fehlerkosten etwa 80% bis 90% der gesamten Qualitätskosten ausmachen und die Kosten für das Auffinden (und Beseitigen) von Fehlern in den Phasen Entwurf, Realisierung, Systemtest und Betrieb im Verhältnis zu 1:8:15:60 stehen, also geradezu exponentiell ansteigen.[57]

Aus dem IT-Pflichtenheft sind nun die **Spezifikationen** als Aufgabenstellungen für die Programmierung auszuarbeiten (IT-Feinkonzept).[58] Dies ist typische Aufgabe des Anbieters.

Ziel muss es sein, die IT optimal an die vorgegebenen fachlichen Anforderungen des Unternehmens anzupassen, um dessen Wettbewerbsfähigkeit zu sichern und möglichst zu verbessern („IT Alignment“).[59] Dies bedeutet freilich nicht, den IT-Einsatz auf das Nötigste zu beschränken und Projekte zur Neu- und Weiterentwicklung radikal zusammenzustreichen, da hierdurch nicht die langfristigen, strategischen Ziele des Unternehmens unterstützt werden.[60] Allgemein müssen die technischen und organisatorischen Strukturen der IT die Unternehmensziele optimal unterstützen und die geplante, künftige Entwicklung des Unternehmens fördern („IT-Governance“).[61] Diese Anforderungen werden wie folgt aufgeschlüsselt:[62]

- Die Leistungserstellung der IT muss im Rahmen der IT-**Security**-Anforderungen (Integrität, Vertraulichkeit, Verfügbarkeit) gewährleistet werden.
- Die IT-Strukturen sollten **offen** und herstellerunabhängig sein.
- Die IT-Strukturen müssen von den Kapazitäten her **skalierbar** sein; die Integration von weiteren Partnern oder Tochterunternehmen muss unkompliziert möglich sein.
- Eine **einheitliche Datenbasis** muss für alle Anwendungen gegeben sein (keine Redundanzen).
- Ein effektives **Schnittstellenmanagement** hat zu erfolgen, d.h., Schnittstellen sind leicht einzubinden und die Verfügbarkeit der Schnittstellen garantiert.
- Die Gesamtkosten (Total Cost of Ownership, TCO) für die einzelnen Anwendungen müssen möglichst gering sein.

---

[57] *Streitz*, IT-Projekte retten, 43.
[58] *Streitz*, IT-Projekte retten, 23.
[59] *Wintersteiger*, IT-Strategien entwickeln und umsetzen, in: Tiemeyer (Hrg.), Handbuch IT-Management, 45.
[60] *Buchta*/*Eul*/*Schulte-Croonenberg*, Strategisches IT-Management, 15.
[61] *Seibold*, IT-Risikomanagement, 164.
[62] *Seibold*, IT-Risikomanagement, 164.

- Eine optimale Geschäftsprozessunterstützung über Unternehmensgrenzen hinweg wird erwartet.

Sorgfältig ist zu prüfen, ob vorhandene ältere Systeme („EDV-Inseln“) wirklich noch weiterhin genutzt werden müssen. Meist müssen hier individuelle Konvertierungsprogramme geschrieben werden sollten, deren Erstellung mitunter mehr kostet als eine sonst in Betracht kommende Anpassung von Standard-Applikationen. Systemeinführungen und –anpassungen sollten deshalb vom Kunden genutzt werden, um Hardware, Applikationen, Daten und Prozesse zu **konsolidieren**.[63]

Bereits bei Vertragsschluss muss geklärt sein, ob der Anbieter für die gesamte geplabte wirtschaftliche Nutzungsdauer ausreichend qualifizierte Hardware-Wartungs- bzw. Software-**Pflegeleistungen** vorhalten und zu marktüblichen Konditionen anbieten wird. Der zu beauftragende Anbieter muss deshalb für diesen gesamten Zeitraum seine vorzuhaltende qualifizierte Leistungsbereitschaft im Systemvertrag garantieren (und notfalls durch Erfüllungsbürgschaften sichern).

Checkliste zur Erstellung des IT-Pflichtenhefts

Die Leistungsbeschreibung bzw. das IT-bezogene Pflichtenheft muss festlegen, [64]

- die vom Auftragnehmer zu erbringenden Leistungen,
- die für Abnahmefähigkeit zu erfüllenden Leistungsmerkmale,
- die Reihenfolge der Projektstufen mit Zeitplan.

Diese Inhalte lassen sich in folgender Weise aufgliedern:[65]

- Feinspezifikation der Funktionen,
- Kosten-/Nutzenschätzung,
- Projektwertanalyse,
- Festlegung der Teilprojekte, Arbeitspakete, Unterlieferanten,
- Verfeinerung der Projektpläne,
- endgültige Leistungsbeschreibung.

---

[63] *Tiemeyer*, IT-Architekturen – planen und managen, in: Tiemeyer (Hrg.), Handbuch IT-Management, 98ff.

[64] Teilweise nach Schneider/v.Westphalen/*Witzel*, Software-Erstellungsverträge, F Rdn. 139.

[65] *Müller-Hengstenberg*, Der Vertrag als Mittel des Risikomanagements, CR 2005, 385, 389.

2. Abschluss des Projektvertrags

Der IT-Projektvertrag muss geeignete Mittel – wie ein regelmäßiges Reporting, Teilabnahmen, Vergütung nach Milestones etc. – vorsehen, die eine **zeitnahe** regelmäßige **Kontrolle** dieser Prüfpunkte erlauben. Diese Fortschrittsüberwachung kann etwa durch Terminlisten (oder Balkendiagramme) durchgeführt werden, die im Intranet geführt und durch taggenaue Eintragung des Projektleiters aktuell gehalten werden, den jeweiligen Soll- und Ist-Status aufzeigen (geplante und tatsächlich erreichte Termine) und von der Geschäftsleitung online jederzeit abgefragt werden können.

Der Kunde sollte nach Möglichkeit alle Leistungen aus einer Hand beziehen. Schließt er getrennte Verträge mit verschiedenen Anbietern, trägt er das volle Risiko, deren Leisrungen zu koordinieren.

Werden mehrere **Nutzungsrechte** an der Software benötigt, sollten die Anzahl der **Lizenzen** („Lizenz“ als Kurzbezeichnung verstanden im Sinne von urheberrechtlichen Nutzungsrechten) und der jeweilige genaue Zeitpunkt des Beginns der Nutzungsberechtigung festgelegt werden, ebenso die Möglichkeiten und Kosten einer regulären Nachlizenzierung (**Lizenzmanagement**). Zu klären und festzulegen ist auch, auf welche (u.U. automatisierte) Weise zusätzliche Lizenzen in Benutzung genommen werden können und die Möglichkeit. nicht mehr benötigte Lizenzrechte möglichst rasch zu terminieren (etwa durch Kündigung).

Auch die **Herausgabe** bzw. **Hinterlegung** der – dokumentierten (!) – **Sourcen** (Quellformate) der Software müssen in der Verhandlungsphase vereinbart werden, da ein diesbezügliches Nachverhandeln oft teuer kommt bzw. ganz ausgeschlossen wird. Herausgabe bzw. Hinterlegung erledigen sich in den Fällen, in denen etwa eine Webseitenerstellung auf XML-Basis erfolgt, da der generierende Code jderzeit am Bildschirm im Browser eingesehen werden kann. Sie scheitert außerdem, soweit der Anbieter selbst Standardkomponenten Dritter einsetzt und hierfür über keine Sourcen verfügt.

Der Abschluss des Projektvertrages setzt das Projekt in Gang und mit diesem die Haftung des Auftragnehmers für die Erfüllung der von ihm übernommenen Leistungspflichten.

Im abzuschließenden Projektvertrag sollte erkennbar sein, welche Regelungen **individuell ausgehandelt** wurden, also nicht der Kontrolle nach AGB-Recht unterliegen. Dies kann dadurch geschehen, dass vorbereitende Dokumente wie Ausschreibungsunterlagen, Leistungsbeschreibungen oder Protokolle aus Verhandlungen etc. dem Projektvertrag als Anlagen beigefügt werden. Individuell ausgehandelt werden meist der Leistungsumfang, die Leistungstermine, Projektmanagement, Qualitätskriterien und Qualitätssicherung, Abnahmeverfahren (Funktionsprüfung) und Vertragsstrafen. Unverändert bleiben hingegen meist Gewährleistungs- und Haftungsregelungen, Fälligkeitsvereinbarungen, Nutzungsrechte und allgemeine Bestimmungen (etwa zum Gerichtsstand).

Rahmenverträge werden in den Fällen verwendet, in denen meist gleichbleibende Regelungen quasi „vor die Klammer gezogen werden". Solche Rahmenverträge sind in der Regel allgemeine Geschäftsbedingungen. Rahmenverträge stellen aber noch keinen Auftrag dar, sondern müssen grundsätzlich durch Einzelaufträge oder –verträge ausgefüllt werden. Der Abschluss allein eines Rahmenvertrages begründet nämlich noch keine Leistungspflicht und verpflichtet als solcher auch nicht zur Erteilung bestimmter Einzelaufträge. Wird aber mit einem Vertrag unter dem Titel „Rahmenvertrag" eine ständige Geschäftsbeziehung aufgebaut und in ihm die Grundlage hierfür sowie die Abnahme einer Mindestzahl von Waren geregelt, so liegt (entgegen der Vertragsbezeichnung) bereits eine verbindliche Bestellung vor. Normalerweise regeln Rahmenverträge nämlich nur bestimmte Einzelheiten erst künftig abzuschließender Verträge.[66]

3. Dokumentation der Anbieterleistung

Der Anbieter muss die Abläufe und Ergebnisse seiner Erstellungsleistungen dokumentieren. Festzulegen ist, welche Dokumentation der Auftraggeber beanspruchen kann. Hier werden (zumindest für größere Anwendungen) vier Dokumentationstypen unterschieden:[67]

- In der **Anwenderdokumentation** werden Einstieg in die und Nutzung der Anwendung beschrieben; ohne diese Anwenderdokumentation ist meist keine Abnahmeprüfung möglich.
- In der **Systemdokumentation** werden technische Angaben zusammengefasst, die für Betrieb, Wartung und Pflege relevant sind.
- Die **Betriebsdokumentation** enthält für die Überwachung und Aufrechterhaltung des Betriebs benötigte Informationen.
- In der **Installationsdokumentation** sollte der Auftragnehmer den Status nach Installation einschließlich erfolgter Parametrisierung festhalten. Oft wird ein am Bildschirm abrufbares Benutzerhandbuch installiert.

Bei Individualentwicklungen oder Anpassungen vorhandener Software oder von zu erwerbender Standardsoftware ist eine **Entwicklungsdokumentation** jedenfalls dann (auch ohne besondere Vereinbarung) geschuldet, wenn der Auftraggeber (z.B. ein anderes Software-Haus) die Software selbst weiterentwickeln oder zumindest pflegen will.

Besteht eine Verpflichtung des Anbieters zur Herausgabe des Quellformats (Sourcen), muss auch eine **Quellcode-Dokumentation** bzw. Beschreibung mit übergeben werden.[68] Die Dokumentation ist ohne besondere Vereinbarung geschuldet; der Anbieter ist insoweit vorleistungspflichtig.[69]

---

[66] OLG Köln, Urt.v.22.4.1994 – 19 U 145/93, CR 1994, 737 – Rahmenvertrag.
[67] Nach *Streitz*, IT-Projekte retten, 54.
[68] S. etwa *Schneider*/v.Westphalen, Software-Erstellungsverträge, C Rdn. 220.
[69] OLG Karlsruhe, Urt.v.16.8.2002 – 1 U 250/01, JurPC Web-Dok. 123/2003.

Wenn der Anbieter die jeweils geschuldete Dokumentation (insbesondere die Anwenderdokumentation bzw. das Bedienerhandbuch) nicht übergibt bzw. liefert, hat er eine Hauptleistungspflicht teilweise **nichterfüllt**[70] und kann der Kunde einen Teil der Vergütung zurückbehalten. Kann der Kunde eine Software ohne Anwenderdokumentation überhaupt nicht nutzen (z.B. nicht einmal laden), wird sogar vollständige Nichterfüllung anzunehmen sein und die Vergütung voll zurückbehalten werden können.

Die jeweilige Dokumentation muss vom Auftragnehmer mangels abweichender Vereinbarung grundsätzlich erst bei Abschluss der Arbeiten an der Software geliefert werden.[71] Im Rahmen der Qualitätssicherung durch den Anbieter muss dieser die Dokumentation parallel zur Software erstellen und nicht erst nachträglich.[72]

4. Spezifische Probleme der Software-Erstellung

a) Parametrisierung

Durch Parametrisieren werden gewissermaßen Einstellungen an im Programm bereits vorgegebenen „Stellschrauben“ (Parametern) vorgenommen. Man stellt so etwa die Anzahl der lizenzierten Nutzer ein. Hier wurde ein Kaufvertrag mit Nebenleistungen angenommen.[73] Eingriffe in den Programmcode sind nicht erforderlich. Diese Einstellungsarbeiten können dennoch recht umfangreich sein, etwa bei Einführung für unternehmenssteuernder Enterprise Resource Planning Software.[74] Auf diese Anpassungsleistung kann Werkvertragsrecht anzuwenden sein.[75] Diese Werkleistung kann sogar den Schwerpunkt des Vertrages bilden.

Eine schöpferische Programmentwicklung erfolgt beim Einstellen der Parameter nicht (aber sehr wohl beim Erstellen der Software mit solchen Parametern); das Einstellen begründet als solches damit auch keinen Urheberrechtsschutz.

b) Programmieren bei Standardapplikationen

Auch bei Standardsoftware können Zusatzfunktionalitäten individuell entwickelt werden (z.B. bei SAP R/3 mit ABAP4). Derartige Entwicklungsprodukte können im Einzelfall durchaus eigenständig urheberrechtlich schutzfähig sein (also unabhängig vom Schutz der Applikation, auf der sie aufbauen.

---

[70] BGH, Urt.v.4.11.1992 – VIII ZR 165/91, NJW 1993, 461 = CR 1993, 422.

[71] BGH, Urt.v. 20.2.2001 – X ZR 9/99, DB 2001, 1141 = CR 2001, 367.

[72] Ausführlich zur Qualitätsssicherung s. *Koch*, IT-Projektrecht, Rdn. 277.

[73] LG Nürnberg-Fürth, CR 1992, 336, 338; Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn. 109.

[74] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 213.

[75] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn. 110.

Zu prüfen ist aber jeweils, ob Applikationen mit derartigen Erweiterungen noch voll „releasefähig“, d.h. trotz individueller Ergänzungen auch mit Folge-Releases lauffähig sind.

c) Portierung

Die Portierung, also Anpassung eines vorgegebenen Programms an andere Betriebssystem-Plattformen kann meist nur von dessen Anbieter durchgeführt werden, da nur dieser über die Sourcen verfügt. Ein solches Portieren gehört i.d.R. nicht zur bestimmungsgemäßen Benutzung der Software, bedarf also, wenn sie von Drittfirmen durchgeführt werden soll, der Zustimmung des Anbieters.

Das Portieren einer vom Besteller zur Verfügung gestellten Software folgt diese Anpassung Werkvertragsrecht,[76] während bei Überlassung einer nach den Kundenanforderungen angepassten Software Werkliefervertrag angenommen wird.[77]

d) Datenmigration

Sollen im einem IT-Projekt Altdatenbestände übernommen werden, bedarf dies grundsätzlich besonderer Vereinbarung, ebenso das Erstellen von Konvertierungsprogrammen für diesen Zweck. Genau geregelt werden sollte, wer für für die **Erfassung** der Altdatenbestände und deren Migration auf eine andere Plattform zuständig sein soll. Vorgeschlagen wird folgendes **Ablaufschema** für die Datenmigration:[78]

- Migrationsplanung im Rahmen der Projektplanung,
- Erhebung/Verifikation der Ausgangsdatenmodelle und der Datenqualität,
- Planung der Datentransformation,
- Transformation in das Zwischenformat,
- Bereinigung, Abgleich,
- Verifikation von Bereinigung/Abgleich,
- Transformation in die Zielformate,
- Laden in die Testumgebung,
- Laden in Produktivumgebung.

---

[76] BGH, Urt.v. 9.10.2001 – X ZR 58/00, CR 2002, 93 = JurPC Web-Dok. 252/2001 (einen Werkliefervertrag i.S.v. § 651 BGB ablehnend)

[77] BGH, Urt.v. 9.10.2001 – X ZR 58/00, a.a.O.; BGH, Urt.v. 14.7.1993 – VIII ZR 147/92, NJW 1993, 2436f.

[78] Nach *Fischbach*/*Lachmann*/*Winnemuth*, Altdatenmigation wird oft unterschätzt, Computerwoche 10/2002, 26.

e) Projektkosten

Der Kunde muss die voraussehbaren Projektkosten bereits bei Vertragsabschluss für die gesamte vorausgesehene Nutzungsdauer kalkulieren. Zu berücksichtigen sind auch zusätzliche Beratungs-, Anpassungs- vergleichbare Leistungen wie etwa Schulungen.

Wird die Abrechnung nach Aufwand vereinbart, sollten die Vertragsparteien den voraussichtlichen Gesamtaufwand in einem **Kostenvoranschlag** verbindlich festlegen. Der Anbieter sollte dann mit Erreichen der verschiedenen Projektstufen den Kostenvoranschlag fortschreiben und einen laufenden Ist/Soll-Vergleich durchführen.[79] Bei Berechnung der Vergütung nach Aufwand sollte grundsätzlich Abrechnung auf der Basis von Stundenzetteln vereinbart werden, in denen die jeweils erbrachte Tätigkeit stichwortartig knapp (aber doch nachprüfbar) bezeichnet wird, ebenso der beteiligte Mitarbeiter und die Stundenzahl.

Teilvergütungen sollten je nach Projektfortschritt (etwa anknüpfend an Milestones) fällig werden. Hierdurch werden nicht nur Zahlungen verschoben, sondern das Entstehen der Zahlungspflicht verschiebt sich oder entfällt, wenn nämlich der Milestone nicht nachweisbar erreicht ist.[80]

Fehlt eine Vereinbarung zur Vergütung im Projektvertrag, gilt die Vergütung dann als geschuldet, wenn die Leistungserbringung nur gegen Vergütung zu erwarten ist (§ 632 Abs. 1 BGB). Fehlt eine Vereinbarung zur Vergütungshöhe, so gilt die „**taxmäßige**“ **Vergütung** als vereinbart (§ 632 Abs. 2 BGB). Grundsätzlich wird die Vergütung erst mit Abnahme fällig (§ 641 Abs. 1 S. 1 BGB). Der Anbieter kann jedoch für in sich geschlossene und erbrachte Teile des Werkes **Abschlagszahlungen** verlangen (§ 632 a BGB), also etwa für selbständig nutzbare Software-Module,[81] die allerdings im wesentlichen mangelfrei sein müssen.

Der Kunde wird die Vereinbarung von **Festpreisen** zu erreichen versuchen. Festpreise werden meist nur für mehr oder weniger standardisierte Leistungen, etwa ein Customizing angeboten, nicht aber für vom Aufwand her schwer kalkulierbare Vorhaben individueller Software-Erstellung oder etwa Altdatenübernahmen. Allerdings besteht auch bei solchen Festpreisvereinbarung die Möglichkeit einer „**Öffnung**“: Kommen nämlich zu den ursprünglich vereinbarten Leistungen erhebliche, zunächst nicht vorgesehene Zusatzleistungen auf Veranlassen des Bestellers hinzu, begründet dies – unabhängig von einer entsprechenden ergänzenden Einigung – einen zusätzlichen Vergütungsanspruch.[82]

---

[79] *Zahrnt*, Projektmanagement von IT-Verträgen, 25.

[80] *Buchta*/*Eul*/*Schulte-Croonenberg*, Strategisches IT-Management, 60; *Koch*, IT-Projektrecht, Rdn. 113 ff..

[81] Palandt/*Sprau*, Bürgerliches Gesetzbuch, § 632 a, Rdn. 5 (selbständig verwertbares Teilprogramm).

[82] BGH, Urt.v.8.1.2002 – X ZR 6/00, JurPC Web-Dok. 98/2002; *Koch*, IT-Projektrecht, Rdn. 121.

f) Zeitplan für Projektaktivitäten

Für jedes IT-Projekt sollte ein verbinflicher Zeitplan erstellt werden, aus dem die Anbieterleistungen und kundenseitigen Mitwirkungshandlungen in eine zeitlicher Abfolge ablesbar sind. Aus einem Projektzeitplan sollte also für jede Phase erkennbar sein, wann etwas zu tun ist, wer etwas zu tun hat und welche Kosten hierbei höchstens entstehen dürfen.[83]

Bei Terminfestlegungen sollten **Reservezeiten** quasi als Puffer vorgesehen werden, um möglichen technischen Schwierigkeiten, gescheiterte Teilabnahmen etc. Rechnung tragen zu können.[84]

g) Installation

Die Vertragsparteien sollten im Projektvertrag ausdrücklich regeln, welcher Vertragspartner die Installation der Software durchzuführen hat. Zu regeln ist auch die Installation von Updates und neue Versionen der Software, gleich, on es sich um Software-Pflegeleistungen oder Nacherfüllung handelt. Zu klären und zu vereinbaren ist auch, ob automatische Installationsroutinen eingesetzt werden und ob eine Funktionsprüfung nach Installation erfolgt (etwa zur Suche neuer Mängel, wenn im Patch gerügte Mängel beseitigt wurden).

5. Change Management

Viele IT-Projekte werden in ihrer Durchführung ergänzt oder geändert. Zur besseren Kontrolle sollten diese Änderungen und Ergänzungen in einem geregelten Verfahren erfolgen, dem sog. „Change Management“. Zwei Typen von Change Requests sind zu unterscheiden:

Änderungen oder **Ergänzungen** der vereinbarten Leistungen sind grundsätzlich nur bei Vereinbarung anbieterseits **geschuldet** und meist mit Mehraufwendungen für den Anbieter verbunden. Als Mehrvergütung kann der Anbieter diese zusätzlichen Kosten nur berechnen, wenn dies (zumindest stillschweigend, § 632 Abs. 1 BGB) vereinbart war. Mit seinem **Änderungs/Ergänzungsantrag** fordert der Kunde den Anbieter auf, ihm, dem Kunden, ein Angebot über die Durchführung zu machen (invitatio ad offerendum). Der Anbieter wird ein solches Angebot meist in den Fällen machen, in denen der Auftraggeber zur Zahlung einer angemessenen Zusatzvergütung bereit ist, hingegen einen Change ablehnen, wenn die Ausführung des gewünschten Changes technisch nicht oder nur unter unverhältnismäßigem Aufwand möglich wäre oder wenn die Auswirkungen des Changes auf bereits erstellte Leistungsteile (gerade im Programmierungsbereich) nicht oder nur sehr schwer kontrollierbar sind. Schon die **Prüfung** von Changes kann **vergütungspflichtig**, wenn sie mit nicht unbeträchtlichem

---

[83] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 218.
[84] *Streitz*, IT-Projekte retten, 4, 50.

Zusatzaufwand verbunden ist. Dem BGH[85] zufolge muss aber der Anbieter bei Vertragsschluss in gewissem Umfang mit Änderungen oder Erweiterungen rechnen und entsprechende Möglichkeiten vorsehen. Auch nach dem Kammergericht Berlin[86] muss der Anbieter (allerdings bezogen auf einen Fall eines fehlenden Pflichtenhefts) mit gewissen Änderungen rechnen und sie in die Vergütung einpreisen; das gilt auch bei Festpreisvereinbarung.[87] Er darf solche **vorhersehbaren Changes** also **nicht** zusätzlich **berechnen**, was seinen Vergütungsanspruch und Gewinn erheblich reduzieren kann.

Mängelbeseitigungen müssen vom Anbieter ohne zusätzliche Vergütung durchgeführt werden.

Für jeden angeforderten Change sollte die Art des Change sowie die durch ihn ausgelösten Mehrkosten und möglichen Terminsverschiebungen in einem **Change Request-Dokument** festgehalten werden[88], das beide Seiten unterzeichnen. Die Durchführung der Changes erfolgt nach den ITIL- und ISO 20000-Spezifikationen.[89] Vor Durchführung eines Change sind die vorhersehbaren **Auswirkungen** des Change auf andere Teile der Anwendung zu prüfen, sei es in Design, Code, Testplan oder Dokumentation.[90] Der Auftragnehmer muss außerdem prüfen, ob seine Software bei Durchführung des jeweiligen Changes noch releasefähig ist, also in der an alle Kunden ausgelieferten Grundfunktionalität unverändert bleibt, und ob weitere Leistungstermine unverändert eingehalten werden können. Eine AGB-Klausel, nach der sich der Kunde bei Durchführung von Änderungs- bzw. Ergänzungsarbeiten (also auch von Change Requests) nicht mehr auf früher vereinbarte Ausführungsfristen berufen kann, wurde AGB-rechtlich als zulässig angesehen.[91]

Jeder Change muss dokumentiert werden und wie die Hauptleistung **qualitätsgesichert** erfolgen.[92] Das muss auch für Mängelbeseitigungen als Form von Changes gelten, – in der Praxis keineswegs die Regel.[93]

6. Mitwirkung des Auftraggebers

Der Kunde hat bei der Projektdurchführung mitzuwirken (Obliegenheit, im Einzelfall auch Schuldnerpflicht, etwa als Pilotkunde[94]). So hat er etwa die für den Anbieter wesentlichen Installationsvoraussetzungen herzustellen, fachkompetentes Personal zur Mitwirkung bei Anlieferung, Installation, Einweisung und Funktionsprüfung zu stellen

---

[85] BGH, Urt.v. 24.6.1986 – X ZR 16/85, CR 1986, 799.
[86] KG Berlin, Urt.v. 1.6.1990 – 14 U 4238/86, CR 1990, 768, 770.
[87] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn.202.
[88] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn.191.
[89] *Koch*, IT-Projektrecht, Rdn. 155.
[90] *Hindel*/*Hörmann*/*Müller*/*Schmied*, Basiswissen Software-Projektmanagement, 186.
[91] BGH CR 1993, 424 – S-Projekt II.
[92] *Streitz*, IT-Projekte retten, 135.
[93] *Koch*, IT-Projektrecht, Rdn. 162.
[94] *Koch*, IT-Projektrecht, Rdn. 173.

und bezüglich des Eigentums des Anbieters (etwa an dessen Rechnern) Verkehrssicherungspflichten bzw. vertragliche Schutzpflichten zu übernehmen.[95] Die **Kosten** für diese Mitwirkung muss der Kunde vor Projektbeginn realistisch abschätzen und in seine Kalkulation einbeziehen. Sie können zwischen 30% und 50% des des gesamten Projektaufwands betragen.[96]

Erfüllt der Kunde seine Obliegenheit nicht, kann der Anbieter selbst nicht in Leistungsverzug geraten[97], eine Entschädigung verlangen (§ 642 Abs. 1 BGB) und muss er nur für grobe Fahrlässigkeit (und natürlich Vorsatz) haften (§ 300 Abs. 1 BGB). Auch soll er bei Nichtliefern einfache Programmiervorgaben durch den Kunden berechtigt sein, sich auf die Grundprogrammierung zu beschränken und Vergütung so zu verlangen, als hätte er vollständig programmiert.[98]

Projektkritische Mitwirkungshandlungen sollten bereits **im Projektvertrag** oder (zur Entlastung der Vertragsdokumente) in einem getrennten Dokument festgelegt werden, das ausdrücklich im Vertrag als Anhang erwähnt und zum Teil des Vertrags gemacht wird. Für den Anbieter wird eine unzureichende Beschreibung notwendiger Mitwirkungshandlungen spätestens dann misslich, wenn er das Unterbleiben einer erforderlichen Mitwirkungshandlung des Kunden (z.B. Pflichtenhefterstellung, Altdatenübergabe zwecks Konvertierung) rügen will. Der Anbieter kann hier u.U. nicht eindeutig feststellen und in einem Prozess beweisen, zu welchem Zeitpunkt der Kunde welche Mitwirkungsleistung hätte erbringen müssen. Der Anbieter sollte also darauf achten, im Projektvertrag ein möglichst vollständiges und klar aufgeschlüsseltes „Obliegenheitsprogramm“ für den Kunden zu definieren. Aus ihm muss der Kunde erkennen können, wann für ihn welche Pflichten aktuell werden. Dies dient zugleich zur auch vertraglich vom Anbieter geschuldeten Aufklärung und Beratung des Kunden, der z.B. Mitarbeiterschulungen und eine Abklärung mit dem Betriebsrat terminlich abstimmen muss.[99]

Der Anbieter muss die jeweils erforderliche Mitwirkung **rechtzeitig abrufen** (z.B. eine Information über Gegebenheiten im Auftraggeberbereich) und bei erkennbarer Unklarheit oder Unvollständigkeit beim Kunden nachfragen.

Vorgeschlagen wird, dass der Auftragnehmer des Auftraggeber regelmäßig über den Projektfortschritt **unterrichtet**.[100] Zwingend erforderlich sind solche Berichte, wenn der Auftragggeber zu einer bestimmten Mitwirkungshandlung oder einer (Teil-)Abnahme aufzufordern ist.

---

[95] *Koch*, Computer-Vertragsrecht, Rdn. 669.

[96] *Streitz*, IT-Projekte retten, 56.

[97] BGH, Urt.v. 23.1.1996 – X ZR 105/93, CR 1996, 467; BGH, Urt.v.13.7.1988 – VIII ZR 292/87, CR 1989, 102 mit Anm. Köhler, CR 1989, 105 = NJW-RR 1988, 1396; OLG Stuttgart, Urt.v.29.3.1994 – 6 U 203/93, CR 1994, 222 (Leitsatz)

[98] BGH, Urt.v.13.7.1988, a.a.O., 102, 104.

[99] *Koch*, IT-Projektrecht, Rdn. 174.

[100] *Streitz*, IT-Projekte retten, 63.

Übersicht über wichtige **typische** kundenseitige **Mitwirkungshandlungen**:[101]

- Übermitteln aller benötigten Informationen an den Anbieter, etwa über (vorhandene) Systemumgebung, Schnittstellen, Unternehmensabläufe (Geschäftsprozesse), fachliche Anforderungen.
- Mitwirken bei Tests, Stellen von qualifizierten Personal (insbesondere Projektleiter) und Testdaten.
- Schaffen der erforderlichen Installationsvoraussetzungen und Internet-Anbindungen (etwa bei Outsourcing).
- Beschaffen erforderlicher Systemkomponenten, soweit nicht vom Anbieter geschuldet.
- Teilnahme ausreichend qualifizierter Mitarbeiter an Schulungen durch den Anbieter.
- Vorbereiten und Durchführen der Abnahme.
- Umgehende, vollständige Fehlermiteilungen auf Anbieterformularen.
- Gewährleisten von Datenschutz und –sicherheit sowie Know-how-Schutz.

7. Projektabnahme

Die Projektdurchführung folgt grundsätzlich Werkvertragsrecht. Die Werkleistung muss (mit der Folge der Fälligkeit der Vergütung und des Beginns des Laufes der Verjährungsfrist) vom Kunden abgenommen werden, wenn sie im wesentlichen vertragsgerecht erbracht wurde (§ 640 Abs. 1 S. 1 BGB).

Teile der Projektleistungen müssen nur abgenommen werden, wenn sie selbständig prüffähig sind. Am Projektende sollte dann eine Schlussabnahme erfolgen: In einem **Integrationstest** ist festzustellen, ob die verschiedenen Teile bzw. Module der Anwendung miteinander lauffähig sind.[102] Die Wirkung früherer Teilabnahme (Billigung von Leistungsteilen als im wesentlichen vertragsgerecht) kann rückwirkend entfallen, wenn der Integrationstest scheitert oder wenn nachträglich Anpassungen an erbrachten Leistungsteilen durchgeführt werden müssen.

Im Rahmen der Abnahmeprüfung kann es schließlich sinnvoll sein, einen **Lasttest** durchzuführen, mit dem der Durchsatz der Daten mit den Massedaten im Dialogbetrieb etwa hinsichtlich der Antwortzeiten geprüft wird.

Festgestellter Fehler der Anbieterleistung werden oft nach ihrer Schwere klassifiziert:[103]

- Der Fehler **verhindert** die Nutzung der Projektergebnisse. Beispiele: Der Fehler verursacht einen Systemabsturz, also einen Abbruch der Verarbeitung. Eine fehlerhafte Dokumentation verhindert die Nutzung wesentlicher Funktionalitäten.

---

[101] Teilweise nach: *Schmidt*, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 1.5 Rdn. 162; *Koch*, IT-Projektrecht, Rdn. 181.

[102] *Müller-Hengstenberg/Wild*, CR 1991, 327, 329 mit dem Hinweis, dass im Integrationstest meist nur wenige Haupttransaktionen getestet werden können und die noch ausstehenden Testaktivitäten 20% - 40% des gesamten Entwicklungaufwands ausmachen können.

[103] *Streitz*, IT-Projekte retten, 60.

- Der Fehler **behindert** die Nutzung der Projektergebnisse. Beispiele: Zur Fortsetzung der Nutzung muss eine Fehlerumgehung erfolgen.
- Der Fehler hat **keine Auswirkung** auf die Nutzbarkeit der Projektergebnisse. Beispiel: Rechtschreibfehler in der Dokumentation (oder in Quellcode-Kommentaren).

Bei Verhinderung wie bei Behinderung der Nutzung kann nicht mehr von unwesentlichen Mängeln gesprochen werden und besteht keine Abnahmepflicht des Auftraggebers (Umkehrschluss aus § 640 Abs. 1 S. 2 BGB). Bereits bei einzelnen solchen Mängeln kann eine Abnahme nicht mehr verlangt werden.[104]

In vielen Fällen ist schon aus Zeitgründen keine vollständige Abnahmeprüfung sämtlicher Funktionen möglich. Hier wird unter Bezug auf eine **80-20-Regel** geraten, jedenfalls die 20% der Funktionen zu prüfen, die 80% des Produktivbetriebs abdecken.[105] Hierzu müssen freilich die Kernfunktionalitäten in der Abnahmespezifikation komplett erfasst werden. Für die restlichen 80% der Funktionen tritt mangels Kenntnis des Bestellers kein Verlust der Mängelrechte ein (arg. e § 640 Abs. 2 BGB). Jedoch sollte im Projektvertrag (oder im Anhang hierzu) geregelt sein, dass nur 20% der Funktionen geprüft werden (können) und diese in einem **Prüfprotokoll** auch bezeichnet werden. Im Abnahmeprotokoll sollte außerdem vorsorglich (gemäß § 640 Abs. 2 BGB) erklärt werden, dass der Auftraggeber sich seine Mängelrechte bezüglich der nicht geprüften 80% der Funktionen **vorbehält**. Vorsorglich ist auch vertraglich zu regeln, dass hinsichtlich dieser 80% eine mögliche (über § 651 BGB anzuwendende) **kaufmännische** Untersuchungs- und **Rügepflicht** nicht eingreift und für die 20% Kernfunktionen die Prüfliste und die Abnahmetermine auch insoweit verbindlich sind.[106]

Bei Feststellung von **Abweichungen** sollten folgende Merkmale **protokolliert** werden:[107]

- Datum, Zeit
- Beteiligte Personen
- Unterscheiden zwischen Teilabnahme und Endabnahme
- Fehlen der Dokumentation oder von Dokumentationsteilen
- Verwendete Systemumgebung, soweit nicht aus der Abnahmespezifikation zu entnehmen.
- Bezeichnung der getesteten Komponente
- Bezeichnung der Testprozedur mit zugehöriger Abnahmespezifikation
- Beschreibung der Abweichung mit erwartetem und tatsächlichem Ergebnis
- Reproduzierbarkeit des Fehlers

---

[104] Aus der technischen Sicht *Streitz*, IT-Projekte retten, 61.

[105] *Streitz*, IT-Projekte retten, 61. Hier wird an das sog. „Pareto-Prizip“ angeknüpft, nach dem 20% der Fehlerurachen 80 % der Fehlerwirkungen erzeugen (*Liggesmeyer*, Software-Qualität, 15).

[106] *Koch*, IT-Projektrecht, Rdn. 189.

[107] Nach *Streitz*, IT-Projekte retten, 141; *Ellenberger/Müller*, Zweckmäßige Gestaltung von Hardware-, Software- und Projektverträgen, 66.

- Auswirkungen des Fehlers (z.B. Abbruch der Verarbeitung, Inkonsistenz der Datenbank)
- Ausdrucke der Systemangaben bzw. Fehlerbilder
- Aufnahme der festgestellten Fehler in eine (von beiden Seite geführte bzw. kontrollierte) Fehlerdatenbank, deren Eintragungen nach Prioritäten abgearbeitet werden.
- Unterschriften vom Kunden und Anbieter

Festzulegen ist auch das **Abnahmeverfahren**, das meist die Form einer technischen und anwendungsbezogenen Funktionsprüfung hat (z.B. Anlage einer elektronischen Personalakte mit Alias-Namen). Hier müssen die für beide Vertragsparteien verbindlichen Abnahme- bzw. Funktionskriterien im Projektvertrag oder in einer Anlage hierzu definiert sein, ebenso das **Testverfahren** und die **Testdaten** (die der Auftraggeber mangels abweichender Vereinbarung beibringen muss). Den Auftragnehmer kann bezüglich der Auswahl geeigneter Testdaten eine Aufklärungs- und Hinweispflicht treffen.[108]

Anstatt der gemeinsamen Funktionsprüfung wird bei größeren Projekten häufig auch ein **Probebetrieb** als Form der Abnahme vereinbart. Auch diese Abnahmeform bedarf besonderer Vereinbarung, da sonst der Probebetrieb leicht als produktive Nutzung zu beurteilen sein kann, die die Fiktion einer erfolgten Abnahme ohne tatsächliche Billigung begründen würde. Klar und genau müssen hier Dauer und Umfang des Probebetriebs festgelegt werden, da erst nach Abschluss die Pflicht zur Erklärung der Abnahme entstehen darf, ebenso die Art des Tests, die vom Anbieter in ihrem Ergebnis als bindend akzeptiert werden müssen.[109] Das Ende des Probebetriebs sollte durch Erklärung des Auftraggebers festgestellt werden.

Für die Durchführung einer Abnahmeprüfung und erst recht eines Probebetriebs der Software muss dem Auftraggeber bereits ein **Recht zur Nutzung** dieser Software eingeräumt sein. Fehlt eine Vereinbarung für diese testbezogene Nutzung, wird aber meist aus der Vereinbarung einer Abnahme-/Probebetriebsregelung zu schließen sein, dass diese Teil der (jedenfalls bei Vertragsbeginn) bestimmungsgemäßen Benutzung (im Sinne von § 69 d Abs. 1 UrhG) sein sollen. Hierzu wird ein Vervielfältigen zwecks Installation gehören, nicht aber ein Bearbeiten, (Weiter-)Verbreiten oder öffentliches Zugänglichmachen. Vorzuziehen ist freilich eine klarstellende Regelung, welche Nutzungen in der Testphase durchgeführt werden dürfen. Festzulegen ist hier etwa auch die Anzahl der zugelassenen Arbeitsplätze bzw. (gleichzeitigen) Zugriffsrechte im Testbetrieb. Der Regelung bedarf auch das Vorgehen bei **Scheitern** der Abnahme, das zu einem restlosen De-Installieren der Software führt, also etwa auch dem (physikalischen) Löschen der Client-Software auf den Arbeitsplätzen.

Erweist sich das Werk (also insbesondere die Software oder deren Änderung) in der Abnahme/Funktionsprüfung als im wesentlichen vertragsgerecht, **muss** der Auftraggeber das Werk **abnehmen** (§ 640 Abs. 1 S. 1 BGB). Wegen unwesentlicher Mängel darf die Abnahme nicht verweigert werden (§ 640 Abs. 1 S. 2 BGB); freilich

---

[108] *Koch*, IT-Projektrecht, Rdn. 191.

[109] Schneider/v.Westphalen/*Redeker*, Software-Erstellungsverträge, D Rdn.236.

bestehen Nacherfüllungsansprüche fort (§§ 634 Nr. 1, 635 BGB). Dies muss grundsätzlich auch für **Teil**abnahmen gelten. Vorausgesetzt sind ausreichende Prüfmöglichkeiten; zumindest müssen Probeläufe ein zufriedenstellendes Ergebnis erbracht haben.[110]

Die **Abnahme bewirkt**:

- Konkretisierung des abgenommenen Werks,
- Verlust des Rechts auf Neuherstellung, Übergang von Erfüllungs- zu Mängelansprüchen,
- Fälligwerden der Vergütung (§ 641 BGB),
- Übergang der Vergütungsgefahr (§§ 644, 645 BGB),
- Verjährungsbeginn für Mängelrechte (§ 634 a BGB),
- Verlust des Rechts auf Nachbesserung bekannter, nicht vorbehaltener Mängel (§ 640 Abs. 2 BGB),
- Entfallen nichtvorbehaltener Vertragsstrafen (§ 341 Abs. 3 BGB),
- Umkehr der Beweislast für Mängel.

8. Rechte an Programmformaten

Bei **Standardsoftware** schuldet der Anbieter nur die Überlassung des ausführbaren Codes, nicht aber der Sourcen (Quellformate),[111] wenn nichts Abweichendes vereinbart wurde. Bei voll vergüteter Entwicklung von **Individualsoftware** erwirbt der Kunde grundsätzlich alle Nutzungsrechte am gesamten Code, also auch an Sourcen, insbesondere bei geplanter eigener Weitervermarktung.[112] Solche Individualentwicklung ist aber heute eher die **Ausnahme**. Anbieter verwenden (insbesondere bei objektorientierter Programmentwicklung) vielmehr bestimmte Software-„Teile“, z.B. ausführbare Programme, aus anderen Aufträgen weiter (Software-Wiederverwendung). Außerdem nehmen sie an komplexen Softwaresystemen (etwa zur Unternehmenssteuerung) oft nur Anpassungen vor. Ausschließliche Nutzungsrechte wird der Kunde hier nur an den individuell entwickelten oder angepassten Software-Teilen erwerben können, an der jeweiligen Gesamtsoftware aber nur einfache Nutzungsrechte (da der Anbieter seine Software sonst nicht mehr weiterverwerten könnte, sondern immer wieder von Grund auf neu schreiben müsste). [113]

Die Anbieterverpflichtung zur Herausgabe der Sourcen kann, soweit sie überhaupt besteht, in AGB **abbedungen** werden.[114]

Außerdem schuldet der Anbieter auch bei Individualentwicklungen die Herausgabe der Sourcen solange nicht, wie er noch **Pflegeleistungen** oder Mängelbeseitigungen an der

---

[110] OLG Düsseldorf, CR 1986, 1091.

[111] OLG München, Urt.v.16.7.1991 – 25 U 2586/91, CR 1992, 208 (Keine Verpflichtung zur Sourcen-Herausgabe, da im Vertrag nur Objektcode erwähnt).

[112] BGH, Urt.v.16.12.2003 – X ZR 129/01, CR 2004, 490.

[113] *Koch*, IT-Projektrecht, Rdn. 208.

[114] LG Köln, Urt.v. 15.4.2003 – 85 O 15/03, JurPC Web-Dok. 232/2003.

Software zu erbringen hat[115] oder diese weiterentwickeln soll. Bei Ende der Pflege- bzw. Gewährleistungsverpflichtung kann der Quellcode dann herauszugeben zu sein, allerdings nicht in der alten, bei Vertragsunterzeichnung vorhandenen, nicht mehr uneingeschränkt verwendbaren, sondern in der **aktuellen**, durch die Pflege angepassten **Fassung**; dies bedarf aber besonderer Vereinbarung.[116]

9. Service Level Agreements – Vereinbarungen abgestufter Anbieterleistungen

Für die Pflege von Software, aber auch die Wartung von Hardware, etwa Web-Serverrechner, oder die Verfügbarkeitsmerkmale für Internet-Anbindungen , werden oft nach Kenmzahlen abgestufte Service Levels vereinbart. Anhaltspunkte für Regelungen finden sich in den Regelwerken ITIL und ISO 20000-2.

Präzise und transparent sollten Qualität (z.B. Verfügbarkeit) und Leistungszeitpunkte geregelt werden, ebenso die Kontrolle mittels Reporting (Möglichkeit der jederzeitigen kundenseitigen Abfrage über den aktuellen Stand durchgeführter Maßnahmen), das schnelle Beseitigen von Leistungsstörungen (Reaktions- und Wiederherstellungszeiten) und Eskalationsprozesse geregelt werden.[117] Notwendig ist außerdem eine

- klare **Abgrenzung** der verschiedenen Service Levels (Leistungsspezifikation) und der Übergänge zwischen diesen. Für jeden Level ist eine Qualität oder einen Quantitätsstandard als messbare genau zu definieren.
- Festlegung von **Sanktionen** (Vergütungsminderung bei kleinen, Vertragsstrafen/pauschalierter Schadensersatz bei mittleren und fristlose Kündigung bei erheblichen Abweichungen[118]) bei Nichteinhaltung der für die jeweiligen SLs geschuldeten Leistungsmerkmale, z.B. Verfügbarkeiten. Die Sanktionen können entsprechend dem Maß der Abweichung gestaffelt werden, von der relativ kleinen Vergütungsminderung über größere Vertragsstrafen bis zur fristlosen Kündigung und/oder Schadensersatz.[119]

Weiter sind in SLAs zu regeln:[120]

- Termine, Fristen, z.B. für die Beseitigung kritischer Störungen.
- Konditionen (Vergütungen, Vertragsstrafen, s.o.).
- Nachweis der Leistungserbringung durch **Messung** (nachprüfbare Aufzeichnungen über Art und Umfang der Leistungen).
- **Reporting** des Anbieters in regelmäßigen Zeitabständen, inwieweit vereinbarte Leistungsmerkmale eingehalten worden sind.[121]

---

[115] BGH, Urt.v.30.1.1986 – I ZR 242/83, NJW 1987, 1259.

[116] *Koch*, IT-Projektrecht, Rdn. 210.

[117] Schneider/v.Westphalen/*Peter*, Software-Erstellungsverträge, G Rdn. 359, 372.

[118] *Bräutigam*, CR 2004, 248, 251; *Braun*, Die Zulässigkeit von Service Level Agreements, 16; *Hörl*/*Häuser*, CR 2003, 713, 717; *Schreibauer*/*Taraschka*, CR 2003, 557, 561.

[119] *Koch*, IT-Projektrecht, Rdn. 228.

[120] *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 374.

[121] *Schreibauer*/*Taraschka*, CR 2003, 557, 560.

- Zulässige Ausreißerquote.
- Einschwingphase (Anlaufzeit), in der SLA-Kriterien noch nicht anwendbar sind.[122]

Hinzukommen können

- die Verpflichtung zur Auslieferung von Updates (in bestimmten zeitlichen Abständen),
- zulässige Wartungsfenster,
- Hotline,
- Bonus-Regelung für besonders gute Einhaltung der Leistungspflichten.[123]

SLAs sehen vor, dass **Störungsmitteilungen** auf verschiedenen definierten Kompetenzebenen (Service Levels) abgearbeitet bzw. je nach Störungsart und -komplexität auf die nächst höhere Ebene weiterreichen[124] und als Leistungen unterschiedlich abrechnen. Die Leistungen können etwa auf Netzwerke, Web-Hosting, Anwendungssoftware oder Kundenbetreuung/Help Desks bezogen sein.[125] In solchen SLAs muss transparent festgelegt sein, welche Levels angeboten und welche Leistungen auf welchem Level erbracht werden und unter welchen Voraussetzungen und in welcher Zeit spätestens für eine Störungsmeldung auf die nächsthöhere Stufe verwiesen wird. [126]

Für **jeden Level** sind also etwa Rahmenbedingungen der Serviceerbringung (Voraussetzungen, Ausnahmen), Servicequalität (Verfügbarkeit,[127] Zuverlässigkeit, Antwortzeiten zwischen Ein- und Ausgabe, Zeit für die Bearbeitung von Transaktionen, etc.), Messmethodik (Häufigkeit, Methode, Verantwortung) und Reporting (Zeit, Art, Verantwortlichkeit) festzulegen[128], ebenso Arbeits- und Bereitschaftszeiten sowie **Reaktionszeiten** (von der Störungs-/Fehlermitteilung bis zum Beginn der Bearbeitung)[129] und eine mittlere Zeitdauer für die Lösung von Benutzerproblemen[130] bzw. zulässige Werte für ein Degrading des Service, für die Dauer von Datenübertragungen, die Übertragungsgeschwindigkeit, der Datendurchsatz/Zeiteinheit und die Antwortzeit zwischen Eingabe eines Zeichen und seine Wiedergabe auf dem Bildschirm oder die Dauer von Mängelbeseitigungen[131] zu definieren. Für den Kunden ist hierbei freilich letztlich nicht die Stufenfolge als solche entscheidend, sondern nur, ob und wann spätestens sowie unter welchem Kostenaufwand er eine Problemlösung

---

[122] *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 375.

[123] *Schneider*, CR 2004, 241, 246; *Koch*, IT-Projektrecht, Rdn. 228.

[124] *Koch*, Computer-Vertragsrecht Rdn.949.

[125] *Towle*/*Bruggeman*, CRi 2002, 75, 76.

[126] *Koch*, IT-Projektrecht, Rdn. 229.

[127] „Verfügbarkeit“ wird definiert als Verhältnis der Zeit, in der der Dienst verfügbar war, zu der Zeit, in der der Dienst theoretisch hätte verfügbar sein können (*Braun*, Die Zulässigkeit von Service Level Agreements, 9).

[128] *Blöse*/*Pechardscheck*, CR 2002, 785, 787.

[129] Bei Reaktionszeiten ist zu differenzieren, ob die Reaktion im Bereich Hotline, User Help Desk, Support, etc. erfolgt (*Hörl*/*Häuser*, CR 2003, 713, 715).

[130] *Diekmann*/*Pickert*, Computerwoche 46/1991, 61, 62.

[131] *Towle*/*Bruggeman*, CRi 2002, 75, 77.

erwarten kann[132]. Die einzelnen Stufen sollten im Vertrag möglichst klar und vollständig vertraglich definiert sein, ebenso die geschuldeten maximalen Zeiten für die Störungsbeseitigung.[133]

Kunden können selbst Verfahren nutzen, um (im Rahmen eines sog. „User-orientierten Servece-Level-Management" (SLM)) die Verfügbarkeit zu **kontrollieren**: [134]

- Tracing des Netzverkehrs (sog. „Network X-Ray"),
- Screen-Capture (Messung am Arbeitsplatz),
- Record/Replay (Messung anhand definierter Workloads) und
- Instrumentierung der Applikation mit Agenten[135].

Zu prüfen ist, ob und welche präventiven („proaktiven") Kontrollen im System möglich sind, etwa laufendes Monitoring von Service Levels, automatische Informationsübergabe an den Help-Desk, Performance- und Verfügbarkeitsprognosen, automatische Fehlerkorrekturen etc.[136]

Festgelegt werden sollten auch die jeweils zu erfüllenden **Mitwirkungsobliegenheiten** des Auftraggebers, so etwa bezüglich des Bereitstellens der Infrastruktur einschließlich Stromversorgung und Klimatisierung von Serverräumen, Zugangsgewährung, Datensicherung.[137]

Die Leistung der Software-Pflege bzw. Systemwartung lässt sich mit verschiedenen technisch definierbaren **Parametern** näher spezifizieren. So bezeichnet man als „Reparaturzeit" (Time To Repair, TTR) die Zeit bzw. die Summe der Zeiten zwischen Ausfall und Neuanlauf des Systems[138]. Die TTR ergibt sich aus der Summe der Diagnosezeit (Detection Time) und der Reparaturzeit. Meist wird eine mittlere Reparaturzeit zugrundegelegt (Mean Time To Repair, MTTR). Gemessen wird weiter die (mittlere) Zeit zwischen zwei Ausfällen ((Mean)Time Between Failures, MTBF)[139]. Für den Anwender ist die Vereinbarung der maximalen Dauer der TTR noch wesentlich wichtiger als die Mindestreaktionszeit, in der mit einer Fehlerbeseitigung begonnen wird; dies gilt besonders, wenn die Anwendung unterbrechungssensibel ist (etwa eine Webpräsenz) und jeder Nutzungsausfall zu einem Umsatzentgang führt. Werden Fehlerbehebungszeiten vereinbart, wird oft eine Unterscheidung nach Fehlerkategorien getroffen, z.B. in „critical", „major" und „minor".[140]

---

[132] *Koch*, Computer-Vertragsrecht, Rdn. 950.

[133] *Koch*, IT-Projektrecht, Rdn. 230.

[134] Nach *Wesseler*, Computerwoche extra 5 v. 6.7.2001, 8, 9. Der Autor berichtet von einer Tagung den beherzigenswerten Satz: „Vereinbaren Sie nichts, was Sie nicht messen und verrechnen können".

[135] Etwa im Rahmen der ARM(Application Response Measurement)-Standardisierung der CECMG (Central Europe Computer Measurement Group).

[136] *Wesseler*, Computerwoche extra 5 . 6.7.2001, 8, 9.

[137] *Börner*/*Buhl*/*Hellmich*/*Klett*/*Moos*, Leitfaden IT-Recht, 57.

[138] *Thurner*/*Dal Cin*/*Schneeweiß*, Verläßlichkeitsbewertung komplexer Systeme, Informatik-Spektrum 21 (1998), 318, 319

[139] *Thurner*/*Dal Cin*/*Schneeweiß*, a.a.O., 319

[140] *v. Baum*, CR 2002, 705, 707; *Koch*, IT-Projektrecht, Rdn. 241.

10. Software-Pflege und Hardware-Wartung

a) Updates, Versionen, Upgrades

Für Software-Leistungen ist zu unterscheiden zwischen Mängelbeseitigungen im Rahmen der Gewährleistung, Fehlerbeseitigung als Pflegeleistung und kleinere Erweiterungen der Funktionalität. Mängelbeseitigungen und kleinere Änderungen werden von Anbietern meist als **Updates** ausgeliefert. Neue „**Versionen**" enthalten hingegen (auch) wesentlich neue Funktionsmerkmale, „**Upgrades**" demgegenüber Programmanpassungen für leistungsstärkere Hardware oder Systeme.[141]

Updates erfolgen meist einheitlich gegenüber der Gesamtheit der Kunden (und nicht terminsgebunden), damit die jeweilige Software-Version einheitlich bleibt und weitergepflegt werden kann.

Soweit Fehler unter einer laufenden Gewährleistung als Mängel zu werten sind, kann in dem Zeitraum zwischen Mitteilung und Behebung durch Update der Lauf der **Verjährungsfrist gehemmt** sein (§ 203 BGB).[142]

In größeren Unternehmen kann jedes **Upgrading** von Software zum aufwendigen Projekt werden. Zu prüfen[143] sind die tatsächlichen Kosten einschließlich Einweisungen bzw. Schulungen tausender Mitarbeiter des Kunden (bei Änderungen der Benutzeroberfläche, z.B. SAP GUI) und Mitwirkung bei der Installation, ebenso notwendige Anpassungen (Changes) der vorhandenen Anwendung durch Wegfall bisheriger individueller Ergänzungen bzw. Erweiterungen. Kostenfallen können auch längere Testphasen sein, ebenso die Übernahme der Daten und eine eingeschränkte Nutzung des bisherigen Release bis zum Beginn des Echtbetriebs („Going live") des Upgrades. Die Durchführung des **Migrationsprojekts** muss in **Phasen** erfolgen, etwa von der Vorstudie zur Ist-Situation, Analyse der optimalen Migrationsstrategie mit Einbinden bisher kundenindividueller Anpassungen und des neuen Release, Implementierung, Integrationstest und Produktivstart.[144]

b) Fristen zur Fehlerbeseitigung, Reaktionszeiten

Ein praxiswichtiges Leistungsmerkmal sind kurze Reaktionszeiten zwischen Fehlermeldung und Eintreffen des Servicepersonals des Anbieters. Die Dauer dieser Reaktionszeiten ist in der Regel von der jeweiligen Anwendung abhängig und bedarf entsprechender konkreter Vereinbarung. Dies gilt im übrigen auch für Reparaturen, die nur gegen gesonderte Vergütung erbracht werden müssen. Das Serviceunternehmen darf hierbei seine Leistung nicht von einer vorhergehenden Erklärung des Kunden abhängig machen, die Kosten als zusätzliche zu übernehmen[145]. Die grundsätzlich als vertragliche

---

[141] *Koch*, IT-Projektrecht, Rdn. 254.
[142] *Koch*, IT-Projektrecht, Rdn. 255.
[143] S. die Übersicht in Computerwoche 51/52 2002, 32.
[144] *Kraus*, Computerwoche 8/2005, 32; *Koch*, IT-Projektrecht, Rdn. 259..
[145] LG Heidelberg, Urt.v. 1.9.1986 – O 53/85 KfH I, IuR 1987, S. 108

Nebenpflicht einzustufende Einhaltung einer bestimmten Reaktionszeit, innerhalb deren mit der Leistung, also etwa mit der Störungsbeseitigung begonnen werden muss, kann bei besonderer Vereinbarung selbst zur Hauptpflicht werden, so etwa, wenn der Nachweis ständiger Reaktionsbereitschaft für den Kunden von wesentlicher Bedeutung ist (etwa bei dem Einsatz von software-gestützten Flugsicherungssystemen). Im Einzelfall ist zu prüfen, ob die Einhaltung durch Vereinbarung einer Vertragsstrafe bei Nichteinhaltung gesichert werden sollte. Auch bei vertragsgemäßer Erfüllung der Reaktionspflicht ist aber noch keine rasche Beseitigung eines Fehlers gesichert.

Nur schwierig vorab im Vertrag fixierbar ist die maximal zulässige **Dauer der Fehlerbeseitigung**, da die Fehlerursachen und erforderlichen Beseitigungsmaßnahmen sehr unterschiedlich sein können. Vermeidbare Verzögerungen (etwa, weil Entwickler mit einem neuen Projekt beschäftigt sind), können Ersatzansprüche aus Pflichtverletzung bzw. aus Verzug auslösen. Das generelle Festschreiben maximaler Fehlerbehebungsfristen in Kunden-AGB kann, jedenfalls dann, wenn Fehlerursachen und -auswirkungen (etwa bei Software Dritter) nicht vorhersehbar sind, nach § 307 Abs. 2 Nr. 1 BGB unwirksam sein, da dem Anbieter ein nicht kalkulierbares Leistungsrisiko aufgelastet würde. Zu prüfen ist hier aber, ob nicht im Einzelfall tatsächlich eigentlich zwischen den Vertragsparteien eine Instand**haltungs**verpflichtung vereinbart worden ist, bei der der Anbieter Vorkehrungen schuldet, dass Fehler gar nicht erst auftreten. Insoweit übernimmt der Anbieter auch das Risiko des Auftretens von Fehlern. Eine solche Vereinbarung eines Leistungsinhaltes unterliegt nach § 307 Abs. 3 S. 1 BGB nicht der AGB-rechtlichen Kontrolle. Das Werk „Wartung“ gilt bezüglich der Systeminstandhaltung als mangelhaft, wenn die Anlage in einen Zustand gerät, in dem Störungen zu erwarten sind, und wenn dieser Zustand auf ungenügende Wartungsarbeiten zurückzuführen sind[146]. Gleiches gilt für Software sinngemäß.

Nicht nur bei der ursprünglichen Implementierung der Software, sondern auch bei erforderlichen Pflegemaßnahmen treffen den Kunden fallspezifisch unterschiedliche **Mitwirkungsobliegenheiten**. Der Kunde hat grundsätzlich alle Anstrengungen zu unternehmen, um eine reibungslose Pflege zu ermöglichen und alles zu unterlassen, was diese Maßnahmen erschweren oder unmöglich machen könnte[147]. So muss der Kunde etwa umgehend auftretende Störungen an (Hardware und) Software mitteilen, die aufgetretenen Fehler, soweit machbar, in einer für den Anbieter nachprüfbaren Form dokumentieren, dem Anbieter die zur Durchführung der erforderlichen Maßnahmen notwendige Zeit am System und Zugang zu diesem einräumen und bei der Leistungserbringung, soweit erforderlich, selbst mitwirken (Vorführen der Fehler, Stellen von Personal etc.)[148].

Nicht abschließend geklärt ist, ob Software-Pflegeleistungen **dokumentiert** werden müssen. Für Leistungen zur Hardware-Wartung wird dies bei Fehlen einer entsprechenden Vereinbarung verneint[149]. Für Software-Pflege kann (und muss) das vereinbarungsunabhängige Bestehen einer Dokumentationspflicht jedenfalls dann bejaht

---

[146] OLG München, Urt.v.22.11.1988 – 25 U 5810/86, BB Beilage 10/1990, 9

[147] *Koch*, Computer-Vertragsrecht Rdn. 970.

[148] *Koch*, Computer-Vertragsrecht Rdn. 970.

[149] OLG München, Urt.v. 24.4.1986 – 1 U 5742/85, CR 1988, S.38

werden, wenn Teile der Software weiterentwickelt werden. Insoweit ist wie bei Neuentwicklungen auch eine Dokumentation geschuldet. Aber auch für sonstige Leistungen zur Software-Pflege empfiehlt sich für den Anbieter schon aus Beweisgründen eine Dokumentation der erbrachten Leistungen.

Jedenfalls sollte die entsprechende **Dokumentationspflicht** vertraglich als **Teil geschuldeter Qualitätssicherung** der Software-Pflegeleistung vereinbart werden. Da Fehlerbeseitigungs- und Entwicklungsarbeiten die Qualität der Software verschlechtern können, sind die durchzuführenden Programmänderungen auf ihre möglichen Auswirkungen auf andere Teile des Programmes (bzw. auf andere Komponenten des Systems) hin zu untersuchen. Diese und sonstige Maßnahmen der Qualitätssicherung sind auch für die Software-Pflege durchzuführen und zu dokumentieren.

Die **Vergütung** für die Pflegeleistungen enthält zumeist zwei Komponenten, nämlich eine Pauschale für regelmäßige Kontrollen und das sonstige Instandhalten der Anlage, und eine anlassbedingt berechnete Forderung aus Sonderkosten für Material und Anfahrt, Einsätze außerhalb der üblichen Geschäftszeit oder die Instandsetzung bei Fehlfunktionen, die vom Kunden zu vertreten sind. Die Pauschale für die Pflege wird auch dann fällig wird, wenn keine Leistung im jeweiligen Abrechnungszeitraum erforderlich war[150] bzw. der Kunde auf Wartungsleistungen verzichtet hat[151]. Werden Leistungen nicht in Anspruch genommen, kann also der Kunde keinen Erstattungsanspruch geltend machen, da das vertraglich geschuldete Werk die Erhaltung des möglichst wenig störanfälligen Zustandes ist[152] bzw. das Vorhalten qualifizierter Leistungsbereitschaft[153] ist.

c) Auf Wartungs- und Pflegeleistungen anwendbares Recht, Mängelhaftung

Leistungen aus Software-Pflegeverträgen folgen immer dann **Werkvertragsrecht**, wenn ein individuell definierter Leistungserfolg, etwa eine Fehlerbeseitigung oder eine (regelmäßige) Aktualisierung, geschuldet ist[154], Dienstvertragsrecht hingegen, wenn eine qualifizierte Tätigkeit zu erbringen ist („best effort“[155]). Die Durchführung geschuldeter definierter Änderungen (Changes) unterliegt grundsätzlich Werkvertragsrecht[156], ebenso diejenige von Ergänzungen oder Funktionserweiterungen oder vom Portieren von Software auf andere Plattformen, das zumeist eine (weitgehend) eigenständige Entwicklung darstellt. Die Leistung ist auf einen vertraglich definierten oder zumindest üblichen Status der Funktionsfähigkeit bezogen. Die Wahl der Mittel zur Erhaltung oder Wiederherstellung dieses Status steht dem Anbieter unter Werkvertragsrecht frei. Nimmt der Kunde vertragsgemäß die Kontrollen bzw. sonstigen entsprechenden

---

[150] OLG Koblenz, Urt.v.17.2.1984 – 2 U 1286/82, IuR 1986, 363; OLG Karlsruhe, Urt.v.28.2.1985 – 9 U 102/83, CR 1987, 232; LG Hagen, Urt.v.26.4.1988 – 21 O 159/87, BB Beilage 15/1989, 8.

[151] AG Hamburg, Urt.v.4.10.1988 – 35 C 444/88, CR 1989, 1102f.

[152] AG Hamburg, Urt.v.4.10.1988 a.a.O.

[153] LG Berlin, Urt.v.22.5.2001 – 10 O 483/00, CR 2001, 743.

[154] LG Köln, Urt. v. 11.5.1984 – 90 O 14/84, DV-R 3, 234.

[155] *Lejeune*, K&R 2002, 441, 445

[156] *Hartmann*/*Thier*, CR 1998, 581, 582.

Tätigkeiten selbst vor und unterstützt ihn das Software-Haus nur beratend, wird (insoweit) grundsätzlich nur **Dienstvertragsrecht** anwendbar sein, ebenso ergänzend für das Vorhalten von Dienstleistungen über den gesamten Vertragszeitraum[157].

Auch **Instandhaltung** als Aufrechterhalten eines vertraglich definierten Status der Funktionsfähigkeit eines Programmes (oder Systems) ist ein (wenngleich auf einen Dauerzustand gerichteter) Erfolg im Sinne des Werkvertragsrecht, ebenso das Bereithalten qualifizierten, jederzeit einsatzbereiten Personals, da dieses Vorhalten zwingende Voraussetzung etwa für ein schnelles Wiederherstellen der Funktionsfähigkeit sein kann. Bei Software-Pflegeverträgen kann sich die Vertragsleistung auf das Vorhalten qualifizierter Leistung (Leistungsbereitschaft) reduzieren[158]. Der Charakter der Pflegeverpflichtung als Dauerschuldverhältnis steht der Anwendung von Werkvertragsrecht nicht entgegen. Der werkvertragliche „Erfolg" wird hier, da der Vertrag für eine gewisse Laufzeit abgeschlossen wird, gewissermaßen zeitlich gestreckt und ist durch laufende Kontrolle herzustellen. Ein werkvertraglicher Erfolg muss nicht begriffsnotwendig als zeitlich punktuell zu erbringender verstanden werden[159].

Gerade bei Dauerschuldverhältnissen und zudem bei den für die EDV-Nutzung wesentlichen Kontroll- und Wartungsaufgaben setzt der Kunde besonderes Vertrauen in die Person bestimmter Mitarbeiter des Anbieters und ist eine Übertragung der Leistungspflichten aus dem Wartungsvertrag auf Dritte (**Subunternehmer**) deshalb nur sehr eingeschränkt möglich, wenn der Kunde nicht zustimmt. Die Vereinbarung einer entsprechenden Übertragungsbefugnis in AGB wurde als unwirksam angesehen, wenn sie formularmäßig das Genehmigungserfordernis des § 415 Abs. 1 BGB ersetzen soll[160].

Mangelhaft ist die werkvertragliche Wartungs-/Pflegeleistung, wenn die Anlage/das System aufgrund der unzureichenden Arbeiten in einen störanfälligen Zustand gerät[161]; das gilt auch für Software. Bei vereinbarter Instandhaltung (Erhalten eines definierten Status der Funktionsfähigkeit) stellt bereits das **Auftreten** eines Fehlers einen Leistungsmangel dar. Bei vereinbarter Instandsetzung löst der Fehler (nach seiner kundenseitigen Mitteilung) hingegen überhaupt erst die Leistungspflicht des Anbieters aus. Von Wartungs-/Pflegearbeiten zu unterscheiden sind **Mängelbeseitigungs-maßnahmen**, die im Rahmen der Mängelhaftung für die Erstellung/Überlassung der Software erbracht werden, kostenfrei zu erbringen sind (§§ 635 Abs. 2, 439 Abs. 2 BGB).

---

157 Für die Anwendbarkeit von Dienstvertragsrecht auf sorgfältig und regelmäßig zu erbringende Wartungsleistungen s. OLG Hamm, Urt.v. 22.11.1988 – 25 U 5810/86, CR 1989, 283f.

158 LG Hagen, Urt.v.26.4.1988 – 21 O 159/87, CR 1989, 814.

159 So wurde etwa die (über einen Zeitraum geschuldete) Bauaufsicht werkvertraglich eingestuft (BGH NJW-RR 1998, 1027). Im EDV-Bereich kann das Erhalten oder Wiederherstellen eines möglichst wenig störanfälligen Zustandes ein werkvertraglicher Erfolg sein (Palandt/*Sprau*, 59.A. 2000, Einf.v. § 631, Rdn. 11explizit „auch für Wartung über längere Zeit hinweg"), ebenso KG CR 1986, 772 (Computerwartungsverträge als Werkverträge mit Dauerwirkung); *Müller-Hengstenberg/Graf v. Westphalen*, DV-Projektrecht 1994, 128.

160 OLG Bamberg, Urt. v. 1.10.1985 – 5 U 57/85, DV-R 3, 34 ff.

161 OLG München, Urt.v.22.11.1988 - 25 U 5810/86, CR 1989, 283f, das die Beweislast für den Schaden (z.B. Head crash) , die Pflichtverletzung und deren Kausalität dem Kunden zuordnet.

11. Regelungen zur Projektbeendigung

a) Parallelbetrieb von alter und neuer Anwendung, Beendigungsunterstützung

Oft laufen alte und neue Anwendung eine Zeitlang parallel. Dies ist etwa bei der Konvertierung von Altdaten auf das neue System der Fall, aber auch bei zunächst nur probeweiser Nutzung des neuen Systems bzw. der neuen Anwendung. **Lizenzrechtlich** kann hier zu klären sein, ob die bisherige Software nicht nur zur produktiven Nutzung benutzt werden darf, sondern auch zu dem Zweck etwa einer Datenmigration auf andere Systeme. Nach dem Lizenzvertrag könnte hier der Bereich der zulässigen bestimmungsgemäßen Benutzung überschritten sein.

Um diese Rechtsunsicherheit zu vermeiden, sollte der Anbieter im IT-Projektvertrag zur **Beendigungsunterstützung** verpflichtet werden. Hierzu gehört die Zustimmung zu Nutzungsläufen der Software zum Zweck der Migration zu anderen Systemen bzw. auf andere Plattformen, aber auch Herausgabe kundenseits benötigter spezifischer Datenformate und ausführbarer Programmteile (z.B. Klassenbibliotheken).

b) Pflichten bei Projektabbruch

Auch bei Projektabbruch – etwa durch (jederzeit mögliche) Kündigung durch den Besteller (§ 649 BGB) oder Rücktritt (§§ 634 Nr. 3, 323, 326 Abs. 5 BGB) – bestehen nachvertragliche Pflichten beider Vertragsparteien aus § 280 BGB, deren Umfang nur im Einzelfall (notfalls unter Heranziehen von § 242 BGB) bestimmt werden kann. So muss der Anbieter Unterlagen und Datensicherungen zurückgeben, die er vom Kunden erhalten hat und die dieser weiterhin benötigt. Dritten darf der Anbieter dieses Material auch nachvertraglich nicht zugänglich machen.

12. Kostensenkungen verhandeln[162]

Bei Wechsel von Hardware und insbesondere Endgeräten sollte auf **Standardisierung** und gut verhandelten Einkauf geachtet werden.[163] Der Einsatz neuerer Technologien macht die IT nicht notwendig in jedem Fall leistungsfähiger;[164] entscheidend ist der jeweilige Einsatzzweck (Beispiele: Wechsel zu Duo-Core-Prozessoren bei einfachen Textverarbeitungsanwendungen bringt fast keine Beschleunigung, wenn die Geschwindigkeit der Texteingabe durch den Menschen unverändert bleibt.)

Bei Software ist auf die Möglichkeit der **Wiederverwendbarkeit** von Software(teilen) zu achten (Template-Verfahren), deren Ausschöpfen allerdings zur Konsequenz hat, dass bestimmte Softwareteile (z.B. Klassenbibliotheken) früher für andere Kunden entwickelt und nun im aktuellen Auftrag nur wiederverwendet werden und insoweit

---

[162] Ausführlich s. *Koch*, IT-Projektrecht, Rdn. 355 ff.

[163] *Dietrich*/*Schirra*, IT im Unternehmen, 4.

[164] *Shipton*/*Turtschi*, Entsprechung und Beweglichkeit als Programm wertorientierter IT-Strategie, in: *Dietrich*/*Schirra*, IT im Unternehmen, 13, 22.

naheliegenderweise dem Kunden keine ausschließlichen Nutzungsrechte eingeräumt werden können. Der Kunde wird dann freilich auch nicht den Preis für eine komplett ausschließliche Rechteinräumung bezahlen wollen.

Bei der anstehenden **Verlängerung** von **Verträgen** kann eine Reduzierung der Vergütung durchsetzbar sein, wenn etwa Wartungs-/Pflegeleistungen nicht in dem ursprünglich vorausgesehenen Umfange in Anspruch genommen wurden. Eine Vergütungsreduzierung lässt sich hier u.a. damit begründen, dass auch die Vorhaltekosten des Anbieter (für das Vorhalten qualifizierter Leistung) entsprechend geringer sind. Sinnvoll kann auch das rechtzeitige Einholen von **Parallelangeboten** sein, um den marktüblichen Preis für eine bestimmte Leistung zum aktuellen Zeitpunkt festzustellen. Demgegenüber sind mögliche, zuweilen nicht unbeträchtliche **Migrationskosten** zu berücksichtigen, die die Vergütungsdifferenz übersteigen können.

Meistbegünstigungsklauseln des Vertrages sollten auch in der Laufzeit genutzt werden. Hierdurch lassen sich Vergütungsreduzierungen erreichen, wenn anderen Kunden des Anbieters für die gleiche Leistung eine niedrigere Vergütung angeboten wird. Hier können Kontakte in User Groups für bestimmte Software nützlich sein.

Monatliche Wartungs-/Pflegevergütungen sollten, wenn ein unterschiedlicher Tätigkeitsumfang zu erwarten ist, nicht pauschal, sondern grundsätzlich mit **Stundennachweis** der Aktivitäten abgerechnet werden. Erhöhungen müssen nicht in jeder Höhe akzeptiert werden. Erhöht ein Anbieter nach Abschluss des Wartungsvertrages die Wartungsgebühren um 700 %, ist dem Kunden nach § 242 BGB ein Festhalten am Vertrag nicht zumutbar, sondern besteht insoweit ein Leistungsverweigerungsrecht.[165]

Die Anzahl der tatsächlich genutzten **Lizenzen** sollte genau kontrolliert werden, nicht nur, um unzulässiges zusätzliches Vervielfältigen zu verhindern, sondern auch, um bei Nichtausschöpfen eines gegebenen Rahmens eine Vergütungsreduzierung zumindest bei Vertragsverlängerungen durchsetzen zu können.

Upgrade-Gebühren haben keine Grundlage im Urheberrecht, wenn sie etwa bei einem erforderlichen Hardware-Wechsel für die Systemsoftware berechnet werden, ohne dass für den Kunden eine zusätzliche oder intensivierte Nutzung möglich ist. Allerdings ist zu unterscheiden: Hat der Kunde eine direkte Vertragsbeziehung zum Anbieter der Systemsoftware, kann er aus dem Überlassungsvertrag dennoch verpflichtet sein (nämlich mit rein schuldrechtlicher Wirkung), wobei bei Vorliegen eines Formularvertrages aber zu prüfen sein kann, ob der Kunde durch eine solche Upgrade-Vergütungsregelung i.S.v. § 307 Abs. 2 BGB unangemessen benachteiligt wird. Hat der Kunde hingegen über ein Systemhaus erworben, das die Systemsoftware selbst zugeliefert erhalten hat, und ist die Upgrade-Klausel nicht im Vertrag mit diesem Systemhaus enthalten, kann der Anbieter der Systemsoftware mangels Vertragsbeziehung den Kunden weder aus Urheberrecht noch aus Vertrag wirksam in Anspruch nehmen.

---

165 LG Coburg, Urt.v.29.1.2002 – 22 O 398/01, JurPC Web-Dok. 346/2002

In **Service Level Agreements** sollten die laufenden Gebühren sowie deren Erhöhungen in die Kostenkontrolle und Wirtschaftslichkeitsprüfung einbezogen sein. Auch müssen Kündigungsfristen überwacht werden, ebenso die Anzahl der tatsächlich unterstützten Geräte.[166] Überprüft werden sollte, ob Service Levels mit sehr hohem Verfügbarkeitsgrad tatsächlich benötigt werde („Premium IT").

Ein Wechsel zu neuen **Releases** kann dann unzumutbar sein, wenn hierfür höhere Hardware- und andere Anforderungen erfüllt sein oder vorhandene Anwendungen/proprietäre Erweiterungen (z.B. mittels ABAP 4) kostenaufwendig integriert müssen, ohne dass der Kunde einen Vorteil in der Nutzung zieht. Das Auslaufen der Unterstützung und die Kündigung von Wartungs-/Pflegeverträgen ist nicht frei möglich. Entscheidend ist die vereinbarte Laufzeit des Vertrages. Die Änderung der Produktpolitik eines Anbieters allein rechtfertigt keine außerordentliche Kündigung,[167] ebensowenig die Weigerung eines Kunden, bei Rechnerwechsel eine „Upgrade-Gebühr" in Höhe von (seinerzeit) DM 19.900,00 zu bezahlen.[168] Ordentliche Kündigung bzw. Nichtverlängern des Vertrages ist zulässig; eine Verpflichtung des Anbieters, noch fünf Jahre nach Ende nach Verkauf des letzten Exemplares (sog. „Life Cycle") Wartungsleistungen verfügbar zu halten[169], wurde überwiegend kritisiert.

Ein Wechsel auf **neue Releases**[170] („Punkt-Null-Versionen") sollte zunächst unterbleiben, da diese meist noch sehr fehlerbehaftet sind, und allein schon die laufende Fehlerbeseitigung erhebliche zusätzliche Arbeitszeitkosten verursachen kann, und oft auch alle Anpassungen, Erweiterungen und Schnittstellen mit überarbeitet werden müssen. In größeren Unternehmen sollten Releases vereinheitlicht werden. Hierzu können Verfahren automatischer Software-Verteilung einzusetzen sein.

Proprietäre **Legacy**-Systeme sollten abgelöst werden, wenn ihr Support bzw. ständige individuelle Anpassungen an Updateslaufend hohe Kosten verursacht. Allgemein sollten historisch gewachsene „Anwendungslandschaften" **konsolidiert** und etwa Spezialanwendungen mit niedrigem Beitrag zum Geschäftserfolg durch Standardapplikationen ersetzt werden. **Server**-Konsolidierung zielt darauf ab, Dienste, Applikationen und Datenbanken auf wenige, hochverfügbare und dynamisch rekonfigurierbare Systeme zusammenzuführen.[171] Kostenvorteile und eine Verbesserung der Datenqualität (Vereinheitlichung der Datenformate, Beseitigen von Redundanzen, vereinfachte Zusammenführung) bringt auch eine Konsolidierung von **Datenbanken** durch Zusammenlegen.[172] IT-Konsolidierung ist auch beim **Merger** von Unternehmen erforderlich, da hier IT-Abläufe vereinheitlich werden können und müssen (um etwa

---

[166] *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 362.

[167] Zutreffend *Welker*/*Schmidt*, CR 2002, 873, 874.

[168] OLG Koblenz, Urt.v.27.5.1993 – 5 U 1938/92, CR 1993, 626 = NJW 1993, 3144.

[169] LG Köln, Urt.v.16.10.1997 – 83 O 26/97, CR 1999, 218.

[170] Unter einem „Releasewechsel" wird meist das Wechseln auf eine höhere Version einer Standardsoftware verstanden. Mit dieser Erhöhung werden Fehler entfernt und Leistungsumfang und/oder –fähigkeit erhöht.

[171] *Ehrle*/*Bukowski*, Computerwoche 17/2003, 38.

[172] *Sprott*/*Froeber*, Computerwoche 25/2003, 40.

Inkompatibilitäten zu beseitigen)[173] und sich Synergien ausschöpfen sowie etwa der Einkauf bündeln lassen.[174] IT-Konsolidierung ist als eigenständiges **Projekt** durchzuführen. Hierbei sind unter Berücksichtigung der gesamten Nutzungsdauer Ist- und erwartbare Soll-Kostenstruktur zu vergleichen

Sind im Vertragsverlauf öfter Mängelbeseitigungen am System oder mängelbedingte Updates erfolgt, die jeweils eine **Minderung** gerechtfertigt haben (§§ 441, 638 BGB), so kann eine Vergütungsreduzierung im Umfange dieser Minderung durchsetzbar sein. Gleiches gilt bei Einschränkungen der Verfügbarkeit von Systemen.

Verwirkte **Vertragsstrafen** können als Reduzierung der Vergütung in Anrechnung bzw. Abzug gebracht werden.

Weist die kaufweise überlassene Software bei Übergabe im **Test** (Funktionsprüfung) Mängel auf, ist eine Zurückweisung wegen Vertragswidrigkeit möglich.[175] Dann ist auch keine Annahme als Erfüllung im Sinne von § 363 BGB erfolgt und kann der Kunde bis zur Beseitigung der Vertragswidrigkeit (Mängel) sein Zurückbehaltungsrecht aus § 320 Abs. 1 S. 1 BGB an der zu leistenden Vergütung geltend macht. Bei größeren Projekten kann bereits der Zinsgewinn deutlich zu Buche schlagen. Voraussetzung ist freilich eine Staffelung der Vergütungszahlungen dergestalt, dass bei Übergabe noch eine ausreichend hohe Restzahlung offen bleibt.

Soweit der Anbieter sein **Haftungsrisiko** summenmäßig **begrenzt**, kann es sinnvoll sein, für das nicht abgedeckte Risiko eine Versicherung abzuschließen und mit dem Anbieter die Übernahme eines Teils der jeweils vereinbarten, marktüblichen Versicherungsprämie zu vereinbaren.

Vermieden werden sollten schließlich typische **Kostenfallen**, so etwa[176]

- die schleichende Erweiterung des Projektumfangs. Das Projekt sollte auf Kernanforderungen reduziert werden. „Nice-to-have“-Anforderungen können gestrichen oder zurückgestellt werden.
- überentwickelte Graphic User Interfaces (GUIs). Die GUI-Entwicklung sollte erst erfolgen, wenn ein endgültiges Format feststeht.
- Der Big-Bang-Ansatz des Projekts. Statt eines komplexen Gesamtprojekts sollten mehrere überschaubare Teilprojekte abgegrenzt werden.
- extensives Testen. 100% aller Fehler lassen sich nur mit unverhältnismäßigem Aufwand (wenn überhaupt) feststellen. Sinnvoll erscheint eine Begrenzung auf Kernfunktionalitäten.
- Mehrfachentwicklungen sollten im auftraggebenden Unternehmen (etwa über Projektdatenbanken) erkannt und verhindert werden.

---

[173] Ausf. zur IT-Konsolidierung s. *Tiemeyer* (Hrg.), Handbuch IT-Management, 99.

[174] *Keese/Wehner*, IT-Konsolidierung im Rahmen einer Post-Merger-Integration, in: Dietrich/Schirra, IT im Unternehmen, 201, 213..

[175] Palandt/*Heinrichs*, Bürgerliches Recht, § 433 Rdn. 41 und 47

[176] Nach: ohne Verf. „Schwarze Löcher im Budget“, Computerwoche 36/2004, 26.

Die angeführten Punkte sind auch bei Unternehmensfusionen zu prüfen, bei denen eine „IT-Merger-Integration“ durchzuführen ist, um Kostensynergien in der Wertschöpfungskette zu erreichen.[177] Kosten lassen sich in diesem Zusammenhang etwa durch Hamonisieren von Geschäftsprozessen und Standardisieren der IT-Infrastruktur erreichen.[178] Als Strategien zur Kosteneinsparungen werden genannt:[179]

13. Risiken und Sanierung von IT-Projekten[180]

a) Risiken

Je größer der Entwicklungs- und/oder Anpassungsteil in IT-Projekten ist, desto mehr Risiken bestehen, dass das Projekt Verzögerungen erleidet oder scheitert. Angesichts der Vielgestaltigkeit der Projekte lassen sich diese Risiken nur allgemein und nichtabschließend auflisten. Typische Risiken sind etwa:[181]

- Versäumen einer unabhängigen technischen Bewertung des Projektvorschlages (u.U. zweites Angebot anderer Firma mit Angabe zur technischen Realisierung oder Kurzgutachten einholen).
- Fehlerhafte Termins- und Aufwandsschätzung.
- Unrealistische oder nicht ausreichend geklärte Projektziele.
- Unverbindliche bzw. vom Projektteam nicht akzeptierte Projektpläne.
- Unvollständige Ermittlung und Festlegung von Anforderungen, keine Erstellung von Fachkonzept bzw. Leistungsbeschreibung.
- Kein einheitliches Verständnis der Kundenanforderungen und –erwartungen.
- Festlegen von Aufwand, Zielvorstellungen, Vorgehen und Zeitplan, bevor der Projektinhalt vollständig bekannt ist.
- Fehlendes oder unvollständiges bzw. widersprüchliches Pflichtenheft.
- Fehlende oder unklare Vereinbarung über die Modalitäten der Funktionsprüfung (z.B. vorgängige Schulung bzw. Einweisung, Anwesenheit von Vertretern beider Vertragsparteien und Führen eines beiderseits zu unterzeichnenden Protokolls, Ergänzungsprüfung von Nacherfüllungsarbeiten.
- Festpreisvereinbarungen ohne Anpassungsmöglichkeiten.
- Fehlende Ressourcen beim Projektstart.
- Überschätzen der Leistungsfähigkeit von Standardsoftware, die mehr als erwartet Anpassungen notwendig macht und zu Kosten- und Terminsüberschreitungen führt.

---

[177] *Buchta*/*Eul*/*Schulte-Croonenberg*, Strategisches IT-Management, 72.

[178] *Buchta*/*Eul*/*Schulte-Croonenberg*, Strategisches IT-Management, 145, 152.

[179] Nach *Dietrich*/Schirra, IT im Unternehmen, 66, 67; .

[180] Ausführlich s. *Koch*, IT-Projektrecht, Rdn. 378.

[181] Teilweise nach: *Müller-Hengstenberg*, Risikomanagement in DV-Projekten, CR 1999, 789 f.; *Streitz*, IT-Projekte retten, 70, 163; Schneider/v.Westphalen/*Witzel*, Software-Erstellungsverträge, F Rdn. 101; *Koch*, Typische Risiken der Vertragsgestaltung für EDV-Projekte, WiB 1997, 178; *Hindel*/*Hörmann*/*Müller*/*Schmied*, Basiswissen Software-Projektmanagement, 30.

- Unterschätzung der Komlexität einer benötigten Lösung.
- Risiko zu langer Realisierungsphasen, dass Anforderungen sich zwischenzeitlich (mehrfach) ändern („Moving Targets“) oder neue Releases von Systemsoftware eingeführt werden müssen und hierdurch Anpassungen erforderlich werden.
- Nichtbeachten der Abhängigkeit der Projektphasen voneinander.
- Unzureichende Durchführung/Kontrolle der einzelnen Phasen.
- Fehlen von Änderungsverfahren (Change Management).
- Unzureichende Abnahmeregelungen.
- Einsatz neuer Technologien ohne Referenzprojekte.
- Nicht vorhandenes oder unzureichendes Krisenmanagement.
- Gesamtfestpreis für mehrstufige Projekte mit nicht klar abgegrenztem Leistungsumfang.
- Unklar definierte oder unzureichende kontrollierte Kundenmitwirkung, z.B. unzureichend Schulung von Mitarbeitern oder unzureichende Aufrüstung vorhandener Rechner.
- Abhängigkeit von verzögert oder schlechterfüllenden Subunternehmern.
- Unzureichende Qualifikation der beteiligten Mitarbeiter.
- Unzureichende oder gar fehlende Qualitätssicherung.
- Unzureichende Strategie für die „Business Continuity“ (Aufrechterhalten des Geschäftsbetriebs auch in kritischen Situationen) oder Disaster Recovery (zu der etwa automatische Datenspiegelung gehört).
- Dauer des Parallelbetriebes von Alt- und Neusystem (Altsystem als „Rückgriffs“-Möglichkeit, für das aber entsprechend länger Lizenzrechte und Wartung/Pflege zu kontraktieren sind).
- Kostenmäßig nicht klar kalkulierte Migrationsprojekte (etwa von R/3 zu Folgeplattformen).
- Funktionen outsourcen, um sie nicht verstehen zu müssen.
- Unzureichende Einbeziehung von Mitarbeitern.

Eine Umfrage der Standish Group im Jahr 2000 ergab beispielsweise, dass nur 28 % der Projekte den versprochenen Leistungsumfang zum geplanten Zeitpunkt, im geplanten Budget und mit der spezifizierten Qualität erreichten.[182] Berichtet wird außerdem, dass etwa 46% aller Softwareprojekte die Termine bis zu 24 Monaten und 59% den Finanzrahmen überschreiten, während 25% der Entwicklungen überhaupt scheitern.[183] Kaum Untersuchungen existieren zu der Frage, wie sich solche Projekt-„Runaways“ (R. Glass) auf die Wettbewerbsfähigkeit des Kunden (und letztlich die Marktpositionierung des Anbieters) auswirken.

[182] Zit. nach *Coldewey*, Warum viele Projekte scheitern, Computerwoche 27/2002, 36.
[183] *Müller-Hengstenberg*, Risikomanagement in DV-Projekten, CR 1999, 789, 790 m.w.N.

b) Projektsanierung

Das Sanieren eines IT-Projejtes setzt voraus, dass die Ursache für das drohende Scheitern identifiziert, der verbleibende Gestaltungsfreiraum eingeschätzt und das Mögliche realisiert wird, ohne die Fehler zu wiederholen.[184]
Ursachen des drohenden Scheiterns (nach Streitz die „Schieflage“ des Projekts) sind etwa das Nichterreichen von Zielen, die mangelnde Akzeptanz durch die Benutzer sowie Termins- und Kostenüberschreitungen genannt,[185] wobei freilich vorausgesetzt ist, dass überhaupt konkrete Leistungstemine und (Teil-) Vergütungen vereinbart sind. Fehlen Terminsvereinbarungen, müssen zunächst (angemessene) Leistungsfristen gesetzt werden. Von erheblicher Bedeutung ist aber auch wohl eine unzureichende Kommunikation zwischen IT-Abteilung und Geschäftsleitung.[186]

Die Sanierung eines Projekts sollte mit der Feststellung des erreichten **Projektstatus** beginnen. Dieser ist mit den Vorgaben aus dem Pflichtenheft für den Soll-Status zu vergleichen, wobei die festgestellten Abweichungen nach ihrem Auswirkungsgrad klassifiziert werden sollten.[187] Wurde bereits mit dem **Produktivbetrieb begonnen**, ist meist der Weg zurück zur ursprünglichen Lösung verbaut, da eine Rückmigration der Daten nicht vorgesehen bzw. möglich ist.[188]

Die einzelnen **Projektphasen** sollten zeitlich und inhaltlich klar voneinander **abgegrenzt** und eine Folgephase erst begonnen werden, wenn die Vorphase beendet und abgenommen ist.[189] Auch sollte die jeweilige Teilvergütung erst mit erfolgter Abnahme fällig werden.

Soweit durchsetzbar, sollten **Vertragsstrafen** für das anbieterseitige Überschreiten von Zeitvorgaben vereinbart werden. Hier muss freilich intern sichergestellt werden, dass Verzögerungen nicht durch Kundenwünsche verursacht wurden, die der Anbieter nicht vorhersehen musste.

Zunächst ist der jeweils erreichte **Leistungsstand festzustellen** und gemeinsam verbindlich zu protokollieren. Von diesem aus ist dann zu prüfen, mit welchem (Zusatz-)Aufwand das Projektziel in welcher Zeitdauer erreichbar ist. Der Kunde muss zusätzlich prüfen, ob die festgestellten Leistungsmängel derart gravierend sind, dass eine weitere Leistungserbringung durch diesen Anbieter für den Kunden nicht mehr zumutbar ist. In diesem Fall ist weiter zu prüfen, ob es kostengünstiger ist, eine Drittfirma mit der Weiterentwicklung („auf der Projektruine“) oder gleich mit einer vollständigen Neuentwicklung zu beauftragen; immerhin erfordert die vorgängige

---

[184] *Heuschele*, Sanierung von EDV-Projekten, CR 1988, 591.
[185] *Heuschele*, a.a.O., 592.
[186] *Heuschele*, a.a.O., 592 nennt hier eine „vollkommene Irritation des Managements bezüglich der widersprüchlichen und wenig nachvollziehbaren Aussagen der beteiligten Projektfachleute“ und ein Sichverschleißen in gegenseitigen Schuldzuweisungen.
[187] *Streitz*, IT-Projekte retten, 69.
[188] *Streitz*, IT-Projekte retten, 168.
[189] *Müller-Hengstenberg*, CR 1999, 789, 791

Prüfung des vorhandenen Codes durch eine Drittfirma selbst oft erheblichen Kosten- und Zeitaufwand.

Erscheint die Fortführung der Entwicklung durch den bisherigen Anbieter als vertretbar (oder gibt es keine Mitbewerber), sollte diese vertraglich neu definiert werden. Hier sollten jetzt unbedingt kontrollfähige kleinere **Leistungsabschnitte definiert** und mit Abnahme/Vergütung bzw. Vertragsstrafenverwirkung verknüpft werden (Milestones). Bei diesen Teilabnahme können/sollten Dritte (Sachverständige etc.) hinzugezogen werden. Zugleich ist ein Vorbehalt zu erklären, dass die Wirkung der Teilabnahme entfällt, wenn der Integrationstest fehlschlägt. Bei Stundensatzhonoraren sollte die **Obergrenze** der zulässig in Rechnung zu stellenden Anzahl der Stunden festgelegt sein. Mindestens ein Drittel der Vergütung sollte erst als Schlusszahlung bei Gesamtabnahme/Integrationstest fällig werden. Auch sollte eine Vertragsstrafe für den Fall weiterer Vertragsverletzungen vereinbart werden, aber durch Individualvertrag. Alternativ sollte anbieterseits eine Erfüllungsbürgschaft vorgelegt werden. Weiter ist zu klären, ob der Auftragsumfang – jedenfalls temporär – auf die wichtigsten Teile des Projektes reduziert werden kann, soweit diese trennbar sind. Leistungstermine sollten nach Möglichkeit als Fixtermine vereinbart werden. Ist dieses nicht möglich, lässt sich ersatzweise eine bindende Leistungsfrist vereinbaren, bei deren ergebnislosem Ablauf der Kunde sofort zurücktreten oder Nichterfüllungsansprüche geltend machen kann.

Von der Werkabnahme unabhängige Abschlagszahlungen können auch noch nach Vertragsschluss vereinbart werden (etwa bei längerdauernden Funktionsprüfungen).[190]

Manches Projekt kann durch eine **Reduzierung** des Projektumfangs gerettet werden.[191] Dies kann etwa durch Verzicht auf nicht zwingend erforderliche Anpassungen oder bestimmte organisatorische Veränderungen erreicht werden.[192] Hierzu kann auch ein stärkeres Strukturieren von abzubildenden **Geschäftsprozessen** in Richtung Standards gehören, um individuelle Anpassungen oder Ergänzungsprogrammierungen unnötig zu machen. Auch ein Zentralisieren der **Datenhaltung** ist vorteilhaft. Kostenintensive eigene „EDV-Inseln" sollten möglichst aufgegeben und deren Aufgaben auf Dienstleister übertragen werden. Weiter ist an eine Aufteilung in eigenständige, zeitlich verschiebbare Projekt**teile** zu denken.

Teilweise finden sich in Verträgen rudimentäre Bestimmungen zu einem „**Eskalationsmanagement**". Diese Regelungen helfen dann wenig, wenn nur festgestellt wird, dass die Vertragsparteien „einen fairen Kompromiss herbeiführen" sollen. Das ergibt sich im Grunde schon aus § 242 BGB, wonach der Schuldner die Leistung so bewirken soll, wie Treu und Glauben es mit Rücksicht auf die Verkehrssitte erfordern. Einzelne Anbieter versuchen eine Konkretisierung, indem sie zu berücksichtigende Faktoren benennen, etwa die „Realisierbarkeit innerhalb des gemeinsam verabschiedeten und fortgeschriebenen Zeitplans", den „Grad der Unabweisbarkeit einer Anforderung", die „Verfügbarkeit einer bereits vorhandenen oder schneller realisierbaren und zumutbaren Lösung" bzw. der „Einfluss der fraglichen

---

[190] BGH, Urt.v. 29.1.2002 – X ZR 231/00, JurPC Web-Dok. 168/2002.
[191] *Streitz*, IT-Projekte retten, 170.
[192] *Streitz*, IT-Projekte retten, 170.

Systemeigenschaft auf Wartungsfreundlichkeit und zu erwartende Performance“, wie in einem ERP-Vertrag eines großen Anbieters formuliert wurde. Formuliert sind hier aber eben nur Anhaltspunkte, jedoch weder eine mögliche Konfliktlösung noch auch nur ein Weg zu dieser. Die Vereinbarung eines Eskalationsmanagements (EM) darf insbesondere nicht dazu führen, dass der Kunde an sich bestehende Erfüllungs- und/oder Gewährleistungsansprüche nicht mehr geltend machen kann. Dies widerspräche schon den §§ 307 Abs. 2 Nr. 2 bzw. 309 Nr. 8 a) BGB. Besonders ist zu beachten, dass der Kunde auch nicht durch nachträgliche individualvertragliche Vereinbarung im EM auf diese Rechte verzichtet. Weiter sollte die Regelungskompetenz nicht einseitig beim Anbieter liegen (der sonst festlegen könnte, wann welche Leistung von ihm erbracht wird); vielmehr muss eine tatsächliche Einigung erfolgen. Dies ist im Vertrag bzw. Ergänzungs-/Sanierungsvertrag festzuhalten, da das EM sonst als einseitiges Leistungsbestimmungsrecht des Anbieters nach § 315 BGB ausgelegt werden könnte.

Auch bei Vertragsverlängerungen (etwa von Software-Pflegeverträgen) kann das Einholen mehrere Angebote die Verhandlungsposition des Kunden stärken und ihm helfen, Preisreduzierungen zu erreichen. Allgemein ist ein **Vertragsmanagement** sinnvoll, in dem regelmäßig die jeweiligen Leistungspflichten (im Vergleich zum aktuell bebötigten Leistungsumfang), Vertragslaufzeiten und Ausstiegsklauseln überprüft werden.

II. Leistungsstörungen im Projekt

Typische anbieterseitige Leistungsstörungen sind Verzug und Schlechtleistung. Sie begründen jeweils Kundenrechte, die in der geeigneten Weise geltend zu machen sind. Bei Vertretenmüssen surch den Anbieter kann der Kunde Schadensersatz verlangen.

1. Kundenrechte aus Anbieterverzug

Gerät der Auftragnehmer mit der Erbringung von Projektleistungen in Verzug, stellt dies eine Pflichtverletzung dar (§ 280 Abs. 1 BGB). Der Auftraggeber kann

- die Zahlung einer vereinbarten Vergütung verweigern, wenn er nicht vorleistungspflichtig ist (§ 320 Abs. 1 S. 1 BGB), oder
- vom Vertrag zurücktreten (§ 323 Abs. 1 BGB) und/oder Ersatz des Verzögerungsschadens verlangen (§ 280 Abs. 1 BGB) oder
- nach Ablauf einer gesetzten Frist Schadensersatz **statt** der Leistung. Der Auftragnehmer ist nicht schadensersatzpflichtig, wenn er die Pflichtverletzung nicht zu vertreten hat (§ 280 Abs. 1 S. 2 BGB), wofür er allerdings darlegungs- und beweispflichtig ist. Freilich scheidet Verzug ohnehin aus, wenn kein Vertretenmüssen besteht (§ 286 Abs. 4 BGB).

In IT-Projektverträgen wird die Durchsetzung von Kundenrechten aus Verzug durch den Umstand erschwert, dass nicht selten keine taggenau datierten Leistungsfristen vereinbart sind oder derart zunächst vereinbarte Leistungsfristen durch nachträglich

vereinbarte Leistungsänderungen oder Mängelbeseitigungen entfallen. Im Rahmen des Change Managements sollten deshalb bei Änderungen oder Ergänzungen unbedingt neue Leistungsfristen vereinbart werden bzw. Aufforderungen zur Mängelbeseitigung grundsätzlich mit einer Fristsetzung verbunden werden.

2. Kundenrechte aus Mängeln der Anbieterleistung

IT-Projekte umfassen meist eine Vielzahl von Einzelleistungen, die unterschiedlichen Vertragstypen zuzuordnen sein können, die wiederum die Mängelrechte festlegen. Der Erwerb von Serverrechnern kann Kaufrecht unterliegen oder als Leasing Mietrecht. Die Erstellung oder Änderung von Software, aber etwa auch von Machbarkeitsstudien oder Feinkonzeptionen durch den Anbieter wird Werkvertragsrecht zuzuordnen sein,[193] ebenso das Projektmanagement als getrennt definierte Leistung, die allerdings nicht auf einen bestimmten Zeitpunkt hin (etwa den der Abnahme), sondern während der gesamten Projektdurchführung geschuldet wird.

Die Einheit des Projekts erlaubt freilich nicht, diese und andere projektspezifische Leistungen vertragstypologisch völlig isoliert zu behandeln, sondern es muss gewissermaßen ein übergreifender vertraglicher „Mantel“ gefunden werden, der eine Vererinheitlichung der Rechtsfolgen bei Leistungsstörungen und hier insbesondere bei Leistungsmängeln erlaubt. Dies gelingt etwa durch Annahme eines Typenkombinationsvertrags.

a) Mängelrechte aus Werkvertrag

Der Anbieter muss frei von Sach- und Rechtsmängeln leisten (§ 633 Abs. 1 BGB). Ob ein Mangel vorliegt, bestimmt sich nach der vereinbarten Beschaffenheit (§ 633 Abs. 2 BGB) bzw. ersatzweise nach der vertraglich vorausgesetzten oder jedenfalls gewöhnlichen Verwendung.

Ist die Werkleistung (etwa das IT-bezogene Pflichtenheft oder die Software) mangelhaft, kann der Auftraggeber die **Abnahme** (§ 640 BGB) **verweigern**, da das geschuldete Werk nicht vertragsgemäß ist. Nimmt der Auftraggeber die Leistung dennoch ab, muss er sich seine Bestellerrechte aus § 634 Nr. 1 – 3 BGB einschließlich Schadensersatzansprüchen vorbehalten, wenn er von Mängeln Kenntnis hat (§ 640 Abs. 2 BGB); Ein entsprechender **Vorbehalt** sollte **im Abnahmeprotokoll** ausdrücklich erklärt werden.

Der Kunde darf nicht übersehen, dass die **kaufmännische Rügepflicht** auch bei Werklieferungsverträgen (§§ 381 Abs. 2, 377 HGB) über Individualsoftware besteht,[194] wenngleich sie für EDV-unkundige Kaufleute hinsichtlich des Umfangs der Mängelschilderung auf das dem Laien zumutbare Maß beschränkt ist.

---

[193] Zum Problem aus § 651 s. S. ..

[194] OLG Celle, Urt. v. 8.11.1985 – 11 U 212/84, IUR 1986, 311.

Aus dem Umstand, dass Software-Fehler nur bei der Definition (fehlerhafte Vorgaben), dem Entwurf (Design-Fehler) oder bei dem Codieren auftreten können, nicht aber erst später,[195] wurde zutreffend die Vermutung abgeleitet, dass Software-Fehler bereits bei **Gefahrübergang** (mit Abnahme, §§ 644, 640 BGB) vorhanden sind.[196] Eine Ausnahme kann allerdings für den Software-Teil Dokumentation gelten, in welche Anbieter auch noch nach Übergabe der Programme Fehler einbauen können, wenn sie die Dokumentation erst im nachhinein erstellen.

Der Kunde kann aus nach Abnahme entdeckten und gerügten Mängel im Rahmen der gesetzlichen Mängelhaftung aus Werkvertrag

- **Nacherfüllung** verlangen (§§ 634 Nr. 1, 635 BGB). Der Nacherfüllungsanspruch hat Vorrang vor weiteren Mängelrechten. Der Anbieter ist als Werkunternehmer (anders als beim Kauf) berechtigt, zwischen Mängelbeseitigung und Nachlieferung zu wählen, § 635 Abs. 1 BGB. Das Nacherfüllungsverlangen – und damit Nachlieferung wie Mängelbeseitigung – stellt im Rahmen der Projektdurchführung einen „Change Request" dar, den der Anbieter ohne Kosten für den Auftraggeber durchzuführen hat. Hierdurch kann eine Terminsverzögerung auftreten, die oft eine Anpassung des Projektplans erforderlich macht.

  Auch Maßnahmen zur Beseitigung von Mängeln sind zu **dokumentieren**. Unzumutbar kann die mehrfache **Wiederholung von Mängelbeseitigungsversuchen** für den Kunden sein, wenn die Nachbesserungsversuche mit personellem und zeitlichen Mitwirkungsaufwand auf der Kundenseite verbunden sind.

  Der Anbieter muss alle **Kosten** der Nacherfüllung tragen, so auch alle Nebenkosten (etwa für Transport, Anfahrt, Arbeitszeit und Material[197]), ebenso erforderliche Nebenarbeiten,[198] etwa von Anpassungen des bereits erstellten, an sich mängelfreien Programmcodes, die durch die Mängelbeseitigung erforderlich werden. Hierzu gehören auch Megraufwendungen für (erneut) durchzuführende Tests.

  Der Anbieter kann die Nacherfüllung aber verweigern, wenn sie für ihn mit unverhältnismäßigen Kosten verbunden wäre (§ 635 Abs. 3 BGB). Der Anbieter ist aber – schon aus seiner vertraglichen Leistungstreuepflicht im Projektvertrag – gehalten, nach Möglichkeit eine temporäre oder dauerhafte Umgehungs- oder Ausweichlösung zu realisieren. Auch wird er zu prüfen haben, ob mögliche Schadensersatzansprüche des Kunden bei Rücktritt wegen Verweigerung der Nachbesserung und Projektabbruch zu einer deutlich höheren Kostenbelastung als eine Mängelbeseitigung (oder sogar Neuerstellung) führen können.

---

[195] Dann sind allenfalls Fehler des Datenträgers denkbar.

[196] LG Coburg, Urt. v. 1.8.1984 – 2 O 478/83, IUR 1986, 314

[197] BGH NJW 1979, 2095.

[198] BGHZ 58, 332, 339.

- Der Kunde darf den Mangel **selbst beseitigen** (§§ 634 Nr. 2 1. Alt., 637 BGB) und hierfür vom Anbieter einen Kostenvorschuss (§ 637 Abs. 3 BGB) sowie Ersatz der erforderlichen Aufwendungen zu verlangen (§§ 634 Nr. 2 2.Alt., 636 BGB), außer, der Unternehmer hat die Nacherfüllung zu Recht verweigert (§ 637 Abs. 1 BGB). Diese Variante zur Beseitigung einer Leistungsstörung im Werkvertragsrecht wird aber zumeist nicht zum Tragen kommen, da der Kunde auch bei Vorliegen von Mängeln nicht die Herausgabe verlangen darf und ohne diese zumeist keine Mängelbeseitigung am Programmcode durchführen kann.

- Der Kunde kann vom Vertrag **zurücktreten** (§§ 634 Nr. 3 1. Alt., 636, 323, 326 Abs. 5 BGB). Damit ist das Projekt freilich durch Abbruch beendet und der Kunde kein Stück seiner Problemlösung näher. Bei nur unerheblichen Mängeln ist der Rücktritt ausgeschlossen (§§ 634 Nr. 3, 323 Abs. 1 BGB), die anderen Gewährleistungsrechte bleiben unberührt.

  Scheitert der Anbieter mit der vertraglich geschuldeten Entwicklungsleistung, muss er im Rahmen der rücktrittsbedingten Rückabwicklung auch erhaltene Lizenzgebühren zurückzahlen und darf er nicht seinen Entwicklungsaufwand in Rechnung stellen.[199]

- Der Kunde kann die Vergütung **mindern** (§§ 634 Nr. 3 2. Alt., 638 BGB). Der Anbieter bleibt auch bei Minderung leistungsverpflichtet. Der Kunde muss aber prüfen, ob ihm mit einer Vergütungsminderung geholfen ist. Berechnet das Programm z.B. Brückenstatiken falsch, ist das Minderungsrecht offensichtlich ungeeignet.

- Der Kunde kann **Schadensersatz** verlangen (§§ 634 Nr. 4 1.Alt., 636, 280, 281, 283, 311 a BGB) oder **Ersatz vergeblicher Aufwendungen** (§§ 634 Nr. 4 2.Alt., 284 BGB). Der Kunde muss prüfen, welchen Weg er hier geht. Stützt er seinen Schadensersatzanspruch auf die §§ 634 Nr. 4, 281 BGB, verlangt er Schadensersatz **statt** der Leistung. Folge ist, dass der Besteller keine (weitere) Projektleistung mehr verlangen kann (§ 281 Abs. 4 BGB). Soll das Projekt weiterlaufen können, darf er also nur Schadensersatz aus den §§ 634 Nr. 4, 280 BGB verlangen, also **neben** der Leistung.

b) Kundenrechte aus Kauf

Der Kunde kann bei Anwendbarkeit von Kaufrecht **Nacherfüllung** verlangen (§§ 437 Nr. 1, 439 BGB). Der Kunde kann aus seiner Käuferposition Mangelbeseitigung oder Lieferung einer mangelfreien Sache verlangen (§ 439 Abs. 1 BGB). Der Anbieter hat kein Wahlrecht. Er muss die erforderlichen Aufwendungen für die Nacherfüllung tragen (§ 439 Abs. 2 BGB). Wäre die Nacherfüllung nur mit unverhältnismäßigem Aufwand möglich, kann er sie verweigern (§ 439 Abs. 3 S. 1 BGB).

---

[199] OLG Köln, Urt.v.14.2.2001 – 19 U 176/95, JurPC Web-Dok. 31/2002

Bei Verweigerung, Fehlschlagen oder Unzumutbarkeit der Nacherfüllung (wegen der Nacherfüllungskosten) kann der Kunde

- vom Vertrag **zurücktreten** (§§ 437 Nr. 2 1. Fall, 440, 323, 326 Abs. 5 BGB), wenn eine angemessene Frist zur Nacherfüllung gesetzt wurde abgelaufen ist (§ 323 Abs. 1 BGB) oder die Fristsetzung entbehrlich war (§ 440 BGB).

- den Kaufpreis **mindern** (§§ 437 Nr. 2 2. Fall, 441 BGB).
- Neben Rücktritt und Minderung kann der Kunde **Schadensersatz** (§§ 437 Nr. 3 2.Alt., 440, 280, 281, 283, 311 a BGB) oder **Ersatz vergeblicher Aufwendungen** (§§ 437 Nr. 3 2. Alt., 284 BGB) verlangen. Der Anbieter haftet, da er Mängelfreiheit als Teil der Erfüllungspflicht schuldet, aus den §§ 281, 283 BGB
  - für **Mangelschäden**[200] (etwa Reparaturkosten[201], mangelfeststellungsbezogene Gutachterkosten[202], verbleibender Minderwert[203], Nutzungsausfall während der Reparatur[204] und Gewinnentgang[205], Betriebsstörung,[206] Betriebsausfallschaden durch verzögerte Inbetriebnahme[207] (!))
  - und aus § 280 Abs. 1 BGB für **Mangelfolgeschäden**,[208] also Schäden an anderen Rechtsgütern des Käufers. Schadensersatz statt Leistung geht (wie bisher Schadensersatz wegen Nichterfüllung) auf das positive Interesse, weshalb der Kunde so zu stellen ist, wie er stünde, wenn der Vertrag ordnungsgemäß erfüllt worden wäre.[209]
  - Bei Gattungssachen besteht eine verschuldensunabhängige Einstandspflicht aus übernommenem Beschaffungsrisiko (§§ 280 Abs. 1, 276 BGB).

c) Haftung aus Garantie

Der Anbieter haftet verschuldensunabhängig aus Garantieübernahme für eine Beschaffenheit oder Haltbarkeit (§ 443 BGB), deren Verjährung sich nicht über § 438 BGB bestimmt, sondern über die §§ 195 ff BGB, weshalb Garantiehaftung mit Schluss des Jahres beginnt (§ 195 Abs. 1 BGB), in dem der Anspruch entstanden ist und der Gläubiger Kenntnis erlangt hat oder erlangen musste. Ein Sachmangel liegt vor, wenn

---

[200] Gesetzesbegründung, BT-Drucks. 14/6040, 224

[201] BGHZ 77, 218

[202] BGH NJW 1978, 2241f

[203] BGH, a.a.O.; Palandt/*Heinrichs*, § 276 Rdn. 110

[204] BGH NJW 1978, 2242

[205] BGHZ 77, 218

[206] *Flessner*, Richtlinie und Reform – Die Einpassung der Kaufgewährleistungs-Richtlinie ins deutsche Recht, in: Grundmann/Medicus/Rolland, Kaufgewährleistung, 233, 244

[207] Gesetzesbegründung, BT-Drucks. 14/6040, 224: Anspruch bereits aus § 280 Abs. 1 BGB, unabhängig von Verzugsvoraussetzungen des § 281 Abs. 1 BGB einschließlich Anspruch auf Ersatz von Rechtsverfolgungskosten.

[208] Gesetzesbegründung, BT-Drucks. 14/6040, 224.

[209] *Boerner*, ZIP 2001, 2264, 2272.

die Sache nicht die garantierte Beschaffenheit aufweist (§ 434 Abs. 1 S. 1 BGB). Die garantierte Beschaffenheit entspricht der vereinbarten Beschaffenheit. Der Verkäufer haftet auf Schadensersatz nach § 280 Abs. 1 BGB, wenn die garantierte Beschaffenheit nicht vorliegt. Auf einen Ausschluss der Haftung für eine Garantie kann sich der Verkäufer nicht berufen (§ 444 BGB); dies gilt auch für Individualverträge.[210] Die Garantie im Sinne des § 443 BGB wird als selbständiger Vertrag verstanden, den auch ein Dritter anbieten bzw. abschließen kann.[211]

In Einkaufs-**AGB** unwirksam dem BGH[212] zufolge ist eine Regelung, nach der der Lieferant auch für unverschuldete Rechtsmängel einzustehen hat. Nach § 307 Abs. 2 Nr. 1 BGB könne nämlich eine Schadensersatzpflicht nur auf Verschulden beruhen. Eine verschuldensunabhängige Schadensersatzhaftung komme nur in Betracht, wenn der Verkäufer für die Beschaffenheit der Kaufsache eine Garantie übernommen hat, nicht aber ohne eine solche Verpflichtung allein aus der gesetzlichen Verkäuferhaftung.

---

[210] *Westermann*, NJW 2002, 241, 247.

[211] *Westermann*, NJW 2002, 241, 248.

[212] BGH, Urt.v.5.10.2005 – VIII ZR 16/05, CR 2006, 221.

Teil III Sonderfälle

1. Einführung von ERP (Enterprise Resource Planning)-Software

Die Einführung von ERP-Software stellt ein geradezu klassisches Beispiel für komplexe IT-Projekte dar.

a) Absicherung überprüfbarer schrittweiser Projektdurchführung

Die – zu Recht – gefürchtete Kostenexplosion etwa bei R/3-Implementierungsprojekten durch Beratungsfirmen resultiert zumeist aus nicht rechtzeitig beseitigten Unklarheiten über die Anfangsbedingungen. Anbieter tendieren dazu, das Projekt allein aus der Perspektive ihrer Software und Konzepte zu sehen. Die Ist-Analyse und ein Soll-Konzept müssen deshalb zur verbindlichen Grundlage des Implementierungsprojekts gemacht werden, damit das Projekt nicht die „Bodenhaftung" mit der benötigten Anwendung verliert. Weiter ist das Projekt in **Projektphasen** aufzuteilen, in denen jeweils überprüfbare Leistungsergebnisse erzielt werden. Im Vordergrund stehen hier weniger Neuerstellungs- als Anpassungsarbeiten, z.B. Parametrisierung und zusätzliche Programmierung. Mit der nächsten Phase darf erst begonnen werden, wenn die vorherige Phase durch den Anbieter nachgewisen erfolgreich abgeschlossen wurde.

b)Vermeidung der Übernahme historisch gewachsener Abläufe

Die Integration von einzelentwickelten, historisch gewachsenen, teils undokumentierten Abläufen[213] (oft „EDV-Inseln" genannt) sollte möglichst vermieden werden, da sie zu erheblichem Mehraufwand führen kann, teilweise sogar zu ständigen Fehlerquellen an den Schnittstellen zur Standardsoftware (etwa bei unterschiedlichen, jedes Mal erst zu konvertierenden Datenformaten). Soweit diese Integration aber unverzichtbar erscheint, müssen der mit ihr verbundene Aufwand (etwa auch zur Erstellung erforderlicher Schnittstellen) und die (voraussichtlichen) Auswirkungen auf Antwortzeiten der gesamten Applikation geprüft werden.

c) Vertragsgestaltung

ERP-Verträge sind meist komplex. Oft werden Rahmen- und Einzelverträge kombiniert, teilweise auch Leistungsverzeichnisse und/oder –scheine angefügt. Die Prüfung auf Vollständigkeit, Widerspruchsfreiheit und AGB-rechtlicher Zulässigkeit muss hier grundsätzlich Klausel für Klausel erfolgen. Einzelne Klauseln werden oft individuell ausgehandelt (z.B. über Verfügbarkeiten), während größere Regelungsbereiche (z.B. zu Nutzungsrechten, Haftung, Schutzrechten und anwendbaren Recht) unverändert bleiben und voll weiter AGB-rechtlicher Kontrolle unterliegen.

---

[213] Zum Problem s. *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 399, 401.

Manche Begriffe bleiben in Anbieterverträgen unzureichend geklärt. Hierzu gehört etwa der Begriff des „indirekten Zugriffs“ auf die Programme, also der Zugriff auf Daten mittels Drittsoftware, etwa CRM-Software. Hier ist zu prüfen, ob die Klausel durch Verwendung eines unzureichend definierten Begriffs intransparent und deshalb gemäß § 307 Abs. 1 S. 2 BGB **unwirksam** ist. Dies ist etwa denkbar, wenn sich für den Kunden nicht erkennen lässt, was genau ein indirekter Zugriff ist und wie viele Benutzerlizenzen hierfür erforderlich sind.[214]

e) „Customizing“ als Teil der vertraglichen Projektleistung

(aa) Business Reengineering

Erfahrungsgemäß treten bei ERP-Projekten immer wieder Schwierigkeiten auf, die vom ERP-Softwareanbieter geschuldete Leistung zu definieren und von Beraterleistungen abzugrenzen, wenn gleichzeitig auch Maßnahmen zum Business (Re-)Engineering (BE) durchzuführen sind. Durch solche BE-Maßnahmen wird nämlich häufig die **Ausgangsbasis geändert**, von der aus die Einsatzbedingungen für die ERP-Software festzulegen ist. Die Anpassung der ERP-Software an die Kundenbedürfnisse (Customizing) durch Wahl vorgegebener Parameterwerte (über Eintragung in Tabellen im Bildschirmdialog), Konfigurierung (Auswahl gewünschter Programmbausteine[215] in das Softwarepaket) oder durch ergänzende Programmierung müssen zur Vermeidung von unnötigen Wiederholungen des Customizing auf den Geschäftsprozessen in ihrer „re-engineerten“ Form aufsetzen.

Führt ein Berater zunächst bestimmte BE-Maßnahmen durch, sind die erzielten Ergebnisse genau und vollständig zu **dokumentieren**, da die BE-Maßnahmen die Ausgangsbasis für das Customizing ändern. Dem Anbieter müssen bei seinem Leistungsbeginn derartige BE-Maßnahmen mitgeteilt worden sein (wenn er sie nicht selbst durchführt); entdeckt er sie erst im Prozess der Leistungserbringung, lösen solche Abweichungen in der Regel zusätzlich kostenpflichtige Changes aus. Die Dokumentation des BE-Zielstatus ist die Grundlage für die ERP-Projektimplementierung. Diese Verknüpfung muss nicht nur organisatorisch-softwaretechnisch, sondern auch in den Verträgen mit dem Berater und dem Software-Anbieter tragfähig verknüpft werden. Mängel oder Lücken in der Dokumentation können unmittelbar in die Projektrealisierung durchschlagen. Die Beraterhaftung ist an diese Konstellation anzupassen; sie sollte für die Dauer des IT-Projekts fortbestehen.

(bb) Customizing und Anpassungsprogrammierung

ERP-Software liefert gewissermaßen Raster für Geschäftsprozesse, die sich unternehmensspezifisch ausgestalten lassen. Bei „Customizing“ erfolgen die

---

[214] Zu dem Problemkreis s. *Niemann*, Lizenzmodelle versagen bei indirektem Zugriff, Computerwoche, 22/2003, 14.

[215] d.h. von Modulen bzw. bei objektorientierten Programmen von Komponenten (*Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 298).

Systemeinstellungen allein durch Eintragen von Parametern in eine Vielzahl von Tabellen.[216] Möglich ist aber teilweise auch ergänzende Programmierung.

(cc) Anwendbares Recht

Die Überlassung auch von ERP-Software wird Kaufrecht (jedenfalls in entsprechender Anwendung) folgen, wenn keine oder keine wesentlichen Anpassungen erfolgen.[217] Möglich ist auch die Begründung einer vertraglichen Nebenpflicht zur Einrichtung. Steht hingegen das Erstellen und Einrichten eines individuell definierten Geschäftsprozesses im Vordergrund, wird Werkvertragsrecht auf die Herbeiführung dieses Zielstatus anzuwenden sein. Ähnlich unterliegen auch individuelle Weiterentwicklungen (etwa mit ABAP/4[218] unter R/3) Werkvertragsrecht. Soweit Werkvertragsrecht anwendbar ist, bedarf die jeweilige Leistung der Abnahme. Sinnvollerweise ist diese Abnahme im Systemvertrag als Funktionsprüfung auszuspezifizieren und festzulegen.

2. Outsourcing

a) Grundkonzeption

Als „Outsourcing" wird die Vergabe ursprünglich selbst wahrgenommener Aufgaben eines Unternehmens an andere Unternehmen bezeichnet, als „IT-Outsourcing" die Vergabe von einzelnen oder allen IT-Funktionen an externe Dienstleister.[219] Diese können **durch Skaleneffekte Leistungen kostengünstiger durchführen**. Outsourcing hat sich aus der Nutzung der Rechenzentren beauftragter Anbieter entwickelt.[220] Application Service Providing (ASP) wird als Ausprägung von Outsourcing verstanden.[221] Auf ASP wird nachfolgend (Rdn. ...) eingegangen.

Grundidee hinter dem Outsourcing-Konzept und dessen wesentlicher **Vorteil** ist, dass sich der Auftraggeber auf **Kernkompetenzen** konzentriert und nicht hierzu gehörende Funktionen und Abläufe auf Auftragnehmer auslagert. Auch können technische Neuerungen leichter und weniger aufwendig nutzbar gemacht werden. Jedoch sollte der Auftraggeber mindestens so viel Kompetenz behalten (Retained Organization), dass er den Outsourcing-Auftragnehmer noch kompetent steuern kann,[222] da Outsourcing-Anbieter und allgemein Lieferanten in der Regel kaum Interesse daran haben, nur einige

---

[216] *Wenzel*, Betriebswirtschaftliche Anwendungen des integrierten Systems SAP R/3, 35.

[217] *Koch*, Computer-Vertragsrecht, Rdn. 1092. Typische Standardeinstellungen sind etwa Anlegen der Unternehmensstruktur (Mandant, Gesellschaften, Buchungskreise), Auswahl des Kontenplans, etc (s. CDI, Basissystem, S. 317)

[218] „ABAP" steht für „Advanced Business Application Programming) und ist eine Programmiersprache der 4.Generation (4GL), die mit „Early Prototyping" arbeitet (CDI, Basissystem, S. 131).

[219] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 451 mit dem Hinweis, dass der Begriff „Outsourcing" aus „outside" und „resource" kombiniert wurde.

[220] *Heymann*/*Lensdorf*, Outsourcing-Vertrag, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 5.4 Rdn. 2.

[221] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 452.

[222] *Dietrich*/*Schirra*, IT im Unternehmen, 7.

wenige durchgängige Standards zu implementieren, wird doch an jedem zusätzlichen Programm und an jeder zusätzlichen Schnittstelle verdient.[223] Das gilt erst recht für Auftraggeber, die selbst in Zulieferketten integriert sind, aber auf punktgenaue (just-in-time erfolgende) Produktionssteuerung mittels IT angewiesen sind. Weiterer Vorteil ist eine bessere Steuerbarkeit der IT-Kosten bzw. Senkung dieser Kosten, Abwälzung von Risiken auf den Outsourcing-Anbieter, Einsparung eigener IT-Spezialisten. **Nachteile** sind eine (teilweise irreversible) Abhängigkeit vom Anbieter,[224] der (teilweise) Verzicht auf eigene IT-Kompetenz sowie eine Verschlechterung der Verständigung zwischen Fachabteilungen und Informationsverarbeitung, die in fremden Händen liegt.[225]

Naheliegend ist Outsourcing bei relativ eigenständigen Aufgaben, etwa Auslagerung der **IT-Security** in „Managed Security Services" (z.B. Perimetersicherung, Firewall-Management, Intrusion-Detection-Service, Gateway-Antivirenschutz sowie Bedrohungs- und Patch-Warnungen, Identity- und E-Mail-Service-Management). Das gilt für die Absicherung allgemein der betrieblichen IT und besonders der Webanbindungen, die Angriffstore öffnen können (wobei das Risiko bei Wireless-Komponenten besonders hoch ist). Im Vertrag mit dem Dienstleister sind die Service Levels spezifisch für jeden Security-Bereich zu definieren.

Die **Kostenkalkulation** für Outsourcing-Projekte birgt erhöhte Risiken, da die Kosten für Transaktionen, Transformation und die enge Bindung an den Anbieter deutlich steigen können und Kostenvorteile erst nach einer Anlaufzeit eintreten.[226] Auch die einzelnen Kostenpositionen sollten bereits anlässlich der Ausschreibung vergleichbar formuliert werden (etwa Preis für Mainframe-Betrieb pro Stunde, Help Desk pro Anruf oder pro Anwender im Monat, pro PC im Monat, pro Server (mit 100 GB, 700 GB, 1,5 TB pro Monat). Auch das Erstellen einer Ist- und Soll-Analyse und die Beschreibung (und gar Erstanalyse) vorhandener Geschäftsprozesse sowie das Erarbeiten von Kontrollverfahren und Messkriterien erfordert personellen und zeitlichen sowie kostenmäßigen Aufwand.[227] Zu klären kann auch sein, wieviele Change Request noch von der vereinbarten Vergütung abgedeckt sind und welche Kosten für zusätzliche Change Requests anfallen.

Beim Auslagern von IT-gestützten Organisationsprozessen werden die IT-Systeme und -Netze der auslagernden Organisation und ihres Outsourcing-Dienstleisters in der Regel eng miteinander **verbunden**, so dass Teile von internen Geschäftsprozessen unter Leitung und Kontrolle eines externen Dienstleisters ablaufen. Ebenso findet auf personeller Ebene ein intensiver Kontakt statt. Grundsätzlich Gleiches gilt für die Ausgründung, also die Auslagerung auf ein hierfür gegründetes Tochterunternehmen oder eine Beteiligungsgesellschaft („Inhouse Outsourcing").[228] In beiden Fällen sollte sichergestellt sein, dass ausreichend fachliches Know-how auf den Dienstleister übergehen kann (ohne dass dieser freilich plötzlich zum Marktkonkurrent wird). Je

---

[223] *Dietrich*/Schirra, IT im Unternehmen, 59.
[224] *Hansen*/*Neumann*. Wirtschaftsinformatik 1, 130.
[225] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 451 f.
[226] O.Verf, Outsourcing will gut vorbereitet sein, Computerwoche 38/2004, 34.
[227] *Schott*, Die Kosten des Outsourcing, Computerwoche 17/2006, 34.
[228] *Klotz*/*Dorn*, Vertragsmanagement in der Informationsverarbeitung, 21..

mehr Know-how übergeht (und je geringer die verbleibende IT-Kompetenz ist), desto größer kann die Abhängigkeit vom Anbieter und ein Wechsel zu einem anderen Anbieter manchmal fast unmöglich werden.

b) Formen des Outsourcing

Es können sehr unterschiedliche Aktivitäten durch Outsourcing auf Dritte übertragen werden, so etwa die Entwicklung von speziellen Applikationen, die Auslagerung von von Datensicherungsabläufen auf Auftragnehmer (Storage und Archivierung, Back-up) oder die Ausführung von Aufgaben (z.B. Lohnbuchhaltung).[229] Eine andere Unterscheidung differenziert nach Grundtypen:[230]

- Bei **Plattform-** (oder Rechenzentrums-)**Outsourcing** wird der vollständige Aufgabenumfang des Rechenzentrumsbetriebs (bzw. des Network-Services, PC-Services) vom Outsourcing-Anbieter übernommen.
- Beim **Application Service Providing** (ASP) stellt der ASP-Anbieter Server-, Betriebs- und Anwendungssoftware zur Verfügung, die mit dem Kunden über Kommunikationsnetze verbunden sind.
- Beim **Business Process Outsourcing** werden komplette Geschäftsprozesse (z.B. Logistik, Vertrieb, Buchhaltung) einschließlich der hierfür notwendigen IT-Services ausgelagert. Auch sonstige Prozesse oder einzelne Funktionen können ausgelagert werden (Outtasking).

c) Risiken von Outsourcing-Projekten

Outsourcing-Vorhaben sind nach den Erfahrungen der Praxis besonders komplex und damit in erhöhtem Maße **risikogefährdet**.[231] Es fehlen ausreichende Absicherungen, um die Vertragserfüllung zu kontrollieren bzw. durchzusetzen. Außerdem besteht ein wesentliches Risiko in einer Abhängigkeit vom beauftragten Outsourcing-Anbieter, da eine Rücktrittsentscheidung nur schwer umsetzbar und wenig realistisch ist, – zumal bei freiwilliger Aufgabe von IT-Management-Know-how.[232] Noch gravierender ist das Risiko, wenn auch Produktionssteuerungsprozesse ausgelagert werden, da hier auch Fachkompetenz aufgegeben wird, die teilweise nicht mehr rückholbar ist.

---

[229] *Tiemeyer*, Organisation und Führung im IT-Bereich, in: Tiemeyer (Hrg.), Handbuch IT-Management, 321, 337.

[230] *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 376.

[231] Nach *Tiemeyer*, Organisation und Führung im IT-Bereich, in: Tiemeyer (Hrg.), Handbuch IT-Management, 321, 338 haben nach einer Umfrage 80% aller befragten Unternehmen u.a. mit unzureichenden Maßnahmen, um die Einhaltung von geschuldeten Spezifikationen sicherzustellen.

[232] *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 377.

Durch Outsourcing auszulagernde Geschäftsprozesse sollten vorher von Fehlern und Inkonsistenzen bereinigt, standardisiert und optimiert werden. Es gilt die Grundregel: „Source niemals ein Problem aus.“[233] Außerdem ist nur bei standardisierten Prozessen ein Benchmarking und ein Wettbewerb möglich.

Besondere Sorgfalt bei der vertraglichen Planung und Absicherung ist allein schon angesichts des vielgestaltigen Leistungsbildes unabdingbar. Dies beginnt bereits bei der **Kostenschätzung**, bei der auch die Anlaufkosten für das Outsourcing-Projekt, die Aufwendungen für die Veränderungsprozesse im Hause des Auftraggebers, die erhöhten Preise für künftige Erweiterungs- und Änderungswünsche gegenüber dem Auftragnehmer sowie die laufenden Zahlungen an diesen einzubeziehen[234] und gegen die erwarteten Kosteneinsparungen gegenzurechnen sind. Allgemein müssen die erzielbaren nominellen Einsparungen durch Auslagerung den zusätzlichen Aufwand für Koordination und Kommunikation übersteigen.[235] Mit zunehmender Anpassung an spezifische Bedürfnisse wird die Leistung komplexer und teurer und verlangt vom Auftraggeber höheren Steuerungsaufwand.[236] Der Steuerungs- und Koordinationsaufwand steigt außerdem überproportional an, wenn der Auftraggeber mit mehreren Outsourcing-Anbietern arbeitet (Multisourcing) und zwischen diesen Partnern Schittstellen bestehen[237] bzw. erst hergestellt werden müssen.

Beim Outsourcing werden Arbeits- oder Geschäftsprozesse einer Organisation ganz oder teilweise zu externen Dienstleistern ausgelagert, damit aber auch Teile der oder die Gesamtheit der Personaldatenverarbeitung. Als typische Beispiele nennt das BSI-GS-Handbuch[238] den Betrieb eines Rechenzentrums, einer Applikation, einer Webseite oder des Wachdienstes. [239] Um diese Leistungen adäquat in IT-Veträgen abbilden zu können, darf nicht nur der Zielstatus, sondern muss auch der **Prozess der Auslagerung** detailliert und in kontrollfähigen Schritten **definiert** werden.

---

[233] *Dietrichsweiler*, Kostensenkung in der denzentralen IT-Infrastruktur, in: Dietrich/Schirra, IT im Unternehmen, 139, 153.

[234] *Tiemeyer*, Organisation und Führung im IT-Bereich, in: Tiemeyer (Hrg.), Handbuch IT-Management, 321, 339.

[235] *Gorritz*/*Habermann*, Gezieltes und strukturiertes Outsourcing von IT, in: Dietrich/Schirra, IT im Unternehmen, 235, 245.

[236] *Gorritz*/*Habermann*, a.a.O., 235, 247.

[237] *Gorritz*/*Habermann*, a.a.O., 235, 247.

[238] GS-Handbuch des BSI, B 1.11.

[239] Das GS-Handbuch unterscheidet folgende Formen: *Outsourcing* ist der Oberbegriff. *Tasksourcing* bezeichnet das Auslagern von Teilbereichen. Werden Dienstleistungen mit Bezug zur IT-Sicherheit ausgelagert, wird von *Security Outsourcing* oder *Managed Security Services* gesprochen (z.B. die Auslagerung des Firewall-Betriebs, die Überwachung des Netzes, Virenschutz oder der Betrieb eines Virtual Private Networks (VPN)). Unter *Application Service Provider* (ASP) versteht man einen Dienstleister, der auf seinen eigenen Systemen einzelne Anwendungen oder Software für seine Kunden betreibt (E-Mail, SAP-Anwendungen, Archivierung, Web-Shops, Beschaffung). Auftraggeber und Dienstleister sind dabei über das Internet oder ein VPN miteinander verbunden. Beim *Application Hosting* ist ebenfalls der Betrieb von Anwendungen an einen Dienstleister ausgelagert, jedoch gehören im Gegensatz zum ASP-Modell die Anwendungen noch dem jeweiligen Kunden.

Endet ein Wartungs- oder auch ein Outsourcing-Vertrag, benötigt der Kunde im Rahmen der erforderlichen **Beendigungsunterstützung** seine Daten rechtzeitig vor Vertragsende, um z.B. den Vertragspartner bruchlos wechseln zu können. Die Rechtsprechung bejahte einen entsprechenden **Anspruch auf Rückübertragung** rechtzeitig vor Vertragsende,[240] ohne dessen Ausgestaltung aber näher festzulegen. Im Vertrag sollten deshalb der passende Zeitpunkt und die Durchführung dieser Rückgabe näher spezifiziert werden. Sinnvoll ist außerdem die Vereinbarung einer **Funktionsprüfung**, ob die Daten verarbeitungstauglich sind. Überhaupt noch nicht gerichtlich entschieden und deshalb unbedingt vertraglich klarzustellen ist, ob der Anbieter diese Daten in einem bestimmten von ihm erarbeiteten und verwendeten Datenformat und –modell herausgeben muss. Deren Herausgabe sollte dann unbedingt vereinbart werden, wenn sonst der Kunde diese Daten nicht oder nur mit unverhältnismäßigem Aufwand bearbeiten könnte. Die Formate bzw. Modelle sollten im Vertrag mit spätestem Herausgabezeitpunkt genau bezeichnet werden.

d) Auf Outsourcing-Projekte anwendbares Recht

Dienstvertragsrecht ist für allgemeine Betreuungsleistungen wie Beratung etc. anwendbar, Werkvertragsrecht für alle Leistungen, die auf Erbringung eines Leistungserfolges abzielen (etwa das zielgerichtete Steuern von Geschäftsprozessen)[241], z.B. das Lauffähigmachen von Software, das Installieren eines Update, die Durchführung eines Backup und zielorientierte Supportleistungen. Das **Überlassen** von **Rechnerkapazität** folgt grundsätzlich Mietvertragsrecht.[242] Der jeweilige Vertragstypus muss, wie sonst auch, von den Gegebenheiten des Falles gemäß § 157 BGB bestimmt werden. Ein typisches Beispiel für eine werkvertragliche Leistung ist das Vorhalten eines Ausfall-Rechenzentrums-Service. Der Service zielt hier darauf ab, dass das Rechenzentrum im Notfall im vereinbarten Leistungsumfang einsatzbereit ist. Dies ist typischerweise ein **werkvertraglich** einzuordnender Leistungserfolg. Auch Dateitransfer kann Werkvertragsrecht unterliegen, wenn die Strecke zu minimieren und das Volumen der zu übertragenden Daten zu maximieren ist. Denkbar ist aber auch die Miete von RZ-Leistungen, und zwar immer dann, wenn der Kunde selbst einen Teil des RZ zur Nutzung (teilweise) überlassen erhält.

In Outsourcing-Verträgen ist meist eine Vielzahl von Einzelleistungen zu regeln. Sie können unterschiedlichen Vertragstypen folgen (z.B. Dienstvertragsrecht bei begleitender Beratung ohne Zielvorgabe). Diese Einzelverträge (die oft die Form von Leistungsscheinen haben, aber eigenständige Verträge sind) lassen sich mit einem Rahmenvertrag bündeln, der gewissermaßen die allen Einzelverträgen gemeinsamen Regelungen vor die Klammer zieht (z.B. zu Change Management, Gewährleistung und Haftung).

---

[240] OLG München, Urt. v. 22.4.1999, CR 1999, 484

[241] Für alle *etwa* Heymann, CR 2000, 23, 26.

[242] Zur Zuordnung s. oben S. ...

e) Ausschreibung von Outsourcing als IT-Projekt

Für die Ausschreibung von Outsourcing-Projekten im nichtöffentlichen, d.h. privatwirtschaftlichen Bereich werden folgende vertragsverhandlungsvorbereitende Stufen unterschieden:[243]

- Das Versenden eines **Request for Information** (RfI) basiert auf einer Anforderungsanalyse und soll als erstes Anschreiben Dienstleister über ein Vorhaben des Absenders informieren und Gelegenheit geben, ihr Interesse an der Durchführung des Projekts zu erklären.
- Einen **Request for Proposal** (RfP) erhalten diejenigen Dienstleister, die auf den RfI geantwortet haben. Der RfP soll die Regeln und den Ablauf des Auschreibungsverfahrens sowie die Entscheidungskriterien und Projektdaten wie Leistungsparameter, Standorte, Qualitätsanforderungen und Preiskonzepte mitteilen. Konkrete Vertragverhandlungen werden vorbereitet (Anbahnungsphase, § 311 Abs. 2 Nr. 1 BGB).
- Im Rahmen einer **Due Diligence** werden in der nächsten Phase sämtliche projektrelevanten Informationen und Verträge offengelegt, wodurch besonderes Vertrauen in Anspruch genommen wird. Der Dienstleister kann weitere erforderliche Informationen einholen und seine Leistungs- nd Kostenschätzung konkretisieren.
- Nach Bewertung der Angebote wird in **kompetitiven Vertragsverhandlungen** mit einem oder zwei Finalisten bis zum Vertragsschluss verhandelt.

Die **Auswahl** des Outsourcing-Anbieters sollte u.a. nach folgenden Kriterien erfolgen:

- Vergleichbare, erfolgreiche Referenzprojekte,
- ITIL-basierte Leistungsangebote,
- Standortnähe,
- ausreichende Anzahl kompetenter Mitarbeiter,
- gesicherte Kapitalstärke des Anbieters.

Die Leistungsbeschreibung sollte eine Auflistung aller betroffenen Assets umfassen.

f) Phasen von Outsourcing-Projekten

Bei IT-bezogenen Outsourcing-Projekten werden zwei grundlegende Stufen unterschieden.[244] Der „Überleitungsteil“ („Transition“) regelt den Übergang des IT-Betriebs, der „Leistungsteil“ dann die vom Outsourcing-Anbieter zu erbringenden Leistungen. Beide Teile sind integriert in ein Projekt zu definieren und zu vereinbaren. Besonders regelungskritisch ist die Transition, da hier von einer individuell

---

[243] Nach: *Scheja*/*Schmitt*, CR 2005, 321, 322/323.

[244] *Heymann*/*Lensdorf*, Outsourcing-Vertrag, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 5.4 Rdn. 5.

unterschiedlichen Ausgangsbasis auszugehen ist. Die überzuleitenden Funktionen und die Art der Überleitung sind deshalb genau zu spezifizieren.

Das Outsourcing-**Projekt** selbst wird in der Literatur in vier Phasen aufgeteilt:[245]

1. Bestimmen der Zielsetzung und Vorbereitung der internen Strukturen. Hier sind die Schnittstellen zu beschreiben, Kommunikationsprozesse zu strukturieren und Service Levels festzulegen, Metriken und Berichtsstrukturen einzuführen.
2. Auswahl des externen Partners bzw. Definition der Organisationsstruktur. Neben Preisen und Service Levels müssen Servicequalität und die Fähigkeit zur reibungslosen Gestaltung des Übergangs der entsprechenden IT-Funktionen evaluiert werden. Das Vertragsvolumen muss unter Wahrung von Vorlauffristen erhöht oder verringert werden können.
3. Übergabe der Aufgabe. Übergang des Personals und gegenenenfalls von IT-Infrastruktur und Lizenzen, gesteuert durch Projektmanaement und Reporting.
4. Fortlaufende Kontrolle des Auftragnehmers.

Diese Abstufungen lassen sich in entsprechender Weise natürlich auch bei Vergabe von anderen Projekten verwenden.

Phasen des Outsourcing-Projekts nach dem BSI

Das GS-Handbuch des BSI unterscheidet sieben Phasen von Outsourcing-Vorhaben:

Phase 1: Strategische Planung des Outsourcing-Vorhabens

Bereits bei der strategischen Entscheidung, ob und in welcher Form ein Outsourcing-Vorhaben umgesetzt wird, müssen die sicherheitsrelevanten Gesichtspunkte herausgearbeitet und zum Teil des Pflichtenhefts gemacht werden.

Phase 2: Definition der wesentlichen Sicherheitsanforderungen

Die wesentlichen übergeordneten Sicherheitsanforderungen für das Outsourcing-Vorhaben müssen die Basis für das Ausschreibungsverfahren sein[246].

Phase 3: Auswahl des Outsourcing-Dienstleisters

Der Outsourcing-Dienstleister ist sorgfältig auszuwählen. Anm. des Verfassers: Hier sind zunächst konsistente Auswahlkriterien festzulegen. Die Auswahl selbst sollte in nachvollziehbarer Weise intern begründet und dokumentiert werden.

---

[245] Nach *Gorritz*/*Habermann*, Gezieltes und strukturiertes Outsourcing von IT, in: Dietrich/Schirra, IT im Unternehmen, 235, 256, 257. Zum BSI-Phasenmodell s. S. ...

[246] GS-Handbuch, M 2.251.

Phase 4: Vertragsgestaltung

Auf Basis des Pflichtenheftes müssen mit dem Outsourcing-Dienstleister vertraglich die gewünschten Leistungen inklusive Qualitätsstandards und Fristen im Einklang mit der vorhandenen Gesetzgebung festgelegt werden (meist in der Form von Service Level Agreements (SLA)), ebenso die genauen Modalitäten der Zusammenarbeit sowie Ansprechpartner, Reaktionszeiten, IT-Anbindung, Kontrolle der Leistungen, Ausgestaltung der IT-Sicherheitsvorkehrungen, Umgang mit vertraulichen Informationen, Verwertungsrechte, Weitergabe von Information an Dritte etc.[247].

Phase 5: Erstellung eines IT-Sicherheitskonzepts für den ausgelagerten IT-Verbund

Auftraggeber und Outsourcing-Dienstleister müssen in enger Zusammenarbeiz ein detailliertes Sicherheitskonzept ausarbeiten[248], das ein Notfallvorsorgekonzept enthält[249]. Der Betriebsrat sollte an dieser Ausarbeitung zumindest beratend beteiligt sein (§ 92 BetrVG).

Phase 6: Migrationsphase

Phase 7: Planung und Sicherstellen des laufenden Betriebs

Besonders sicherheitskritisch ist die Migrations- oder Übergangsphase, die deshalb einer sorgfältigen Planung bedarf[250].

g) Abnahme von Outsourcing-Leistungen

Entscheidend für die Festlegung von Abnahmekriterien ist der konkret vereinbarte Leistungsumfang. Hierzu gehören etwa die Verfügbarkeit des Outsourcing-Systems oder bestimmte Systemkomponenten, ebenso die Ausgestaltung des Datentransfer (Streckenminimierung bzw. Volumenmaximierung).

---

[247] GS-Handbuch, M 2.253 Vertragsgestaltung mit dem Outsourcing-Dienstleister.
[248] GS-Handbuch, M 2.254.
[249] GS-Handbuch, M 6.83.
[250] GS-Handbuch, M 2.255.

h) Zentrale Regelungspunkte in Outsourcing-Verträgen[251]

- **Auswahl** der auszulagernden Prozesse bzw. Funktionen. Diese sollten möglichst gut strukturiert und standardisiert sein.
- **Mindestvertragslaufzeit**: Entweder durch Kündigungsregelungen oder durch ausdrückliche Bestimmung der Vertragslaufzeit ist sicherzustellen, für welchen Zeitraum der Auftraggeber gesichert mit der Nutzbarkeit der Leistungen rechnen kann.
- **Marktübliche Preise**: Leistungsabhängiges Preismodell mit Bonus-Malus-System.
- **Definition des Service:** (Name, Ansprechpartner, Beschreibung, Serviceort, etc.), Bezeichnen geschäftskritischer Prozesse, Einhalten gesetzlicher und sonstiger Regelungen sowie einschlägiger Normen.[252]
- **Transition**: Verlagerung der Leistungserbringung an den zukünftigen Ort (Anbieter-RZ)[253], Festlegung der Ausgangsbasis (Ist-Zustand) und Zielvorgabe, Milestones, Vergütung für Transition-Leistungen, Abnahmeregelung.
- **Service Levels**: Typisierte Leistungsumfänge bzw. Leistungsparameter (z.B. Kernservicezeiten, Bereitschaftszeiten, Erreichbarkeit, Selbstlösungs- und Fehlerraten, **Verfügbarkeit** während definierter Zeiträume („Wartungsfenster"), maximal zulässige Ausfallzeiten, Reaktions- und (wenn möglich) Fehlerbeseitigungsfristen, maximale Systemantwortzeiten), jeweils als Mindeststandards, Vorsorgemaßnahmen und Notfallpläne, gestaffelte **Sanktionen** bei Verletzung (z.B. Minderung/pauschalierter Schadensersatz/Kündigung[254] und Vertragsstrafen[255]), ebenso Messverfahren und **Reporting** durch Anbieter über die Einhaltung der Vorgaben in Service Level Agreements vereinbaren.
- **Spezifikation von Schnittstellen**: Sind die IT-Systeme des Auftragebers und des Auftragnehmers inkompatibel, muss in einem Migrationsprojekt vorab eine Anpassung erfolgen.[256] Diese kann bei einer Vielzahl von Komponenten sehr umfangreich ausfallen. Spezifikation von technischen Standards.
- **Sicherheitsregeln**
- **Mandantenfähigkeit** des IT-Systems des Anbieters (Auslegung für mehrere Auftraggeber).

---

[251] Teilweise nach *Lux*/*Schön*, Outsourcing der Datenverarbeitung 1997; *Gadatsch*, IT-Controlling, in: Tiemeyer (Hrg.), Handbuch IT-Management, 359, 397; *Klotz*/*Dorn*, Vertragsmanagement in der Informationsverarbeitung, 20. Die Auflistung kann schon aus Raumgründen nur die wichtigsten Regelungspunkte erfassen. *Heymann*/*Lensdorf*, Outsourcing-Vertrag, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 5.4 Rdn. 12; die Verfasser warnen zutreffend, dass Outsourcing-Verträge i.d.R. sehr umfangreich ausfallen (bei den Autoren erstreckt sich ein Vertragsmuster auf etwa 96 Druckseiten). *Blöse*/*Pechardscheck*, CR 2002, 785, 787.

[252] *Bräutigam*, IT-Outsourcing, Teil 11, Rdn. 23.

[253] *Blöse*/*Pechardscheck*, CR 2002, 785, 788.

[254] *Hörl*/*Häuser*, CR 2003, 713, 717.

[255] Vertragsstrafen werden mangels abweichender Vereinbarung nur bei Verschulden verwirkt (BGH BB 1991, 2252).

[256] *Heymann*/*Lensdorf*, Outsourcing-Vertrag, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 5.4 Rdn. 166.

- Festlegung der **Mitwirkungspflichten** des Auftraggebers einschließlich Bereitstellen technischer Infrastuktur und Gewährung von Zugang.[257]
- Regelmäßiger kundenseitiger Check (z.B. alle zwei Jahre) etwa der kundenspezifischen Ausrichtung der Service Levels und der Einhaltung von Verfügbarkeiten.
- Anpassungs- und Weiterentwicklungspflicht des Anbieters.
- Möglichkeiten der **Leistungsänderung** (Change Management).[258]
- Beauftragung von **Subunternehmern** durch Auftragnehmer nur bei Einwiligung des Auftraggebers.
- Eventuell Übereignung von Hardware des Auftraggebers an den Outsourcing-Anbieter; hier Mängelhaftung des Auftraggebers regeln.
- **Lizenzrecht** des Anbieters zur Nutzung von Software, die dem Kunden (Auftraggeber) von Dritten (Lizenzgeber der Software) überlassen wurde. Stimmt der Lizenzgeber nicht der Übertragung des Nutzungsrechts vom Kunden auf den Anbieter zu, kann die Notwendigkeit entstehen, dass der Anbieter selbst eine Lizenz erwerben muss.
- **Nutzungsrechte** an vom Anbieter entwickelter Software (etwa bei Wechsel des Kunden zu anderem Anbieter zwecks Erhalten der Kontinuität der Datenverarbeitung).
- Wartung/Pflege des Anbietersystems.
- Regelung für Verzug bzw. Teilverzug mit der Leistungserbringung.
- Regelung, unter welchen Voraussetzungen eine wiederholte bzw. eine schwere Pflichtverletzung vorliegt.
- **Mängelhaftung** (soweit Kauf- bzw. Werkvertragsrecht anwendbar) während der gesamten Vertragslaufzeit. Zu unterscheiden sind Mängel der Transition-Leistungen und der eigentlichen Outsourcing-Leistungen. Fehlerklassendefinitionen sind meist nur auf letztere anwendbar.
- **Eskalationsverfahren**
- Regelung der **Übernahme der Software**, wenn die Software des Auftraggebers vom Outsourcing-Anbieter übernommen werden soll. Im Vertrag Auftraggeber – Software-Anbieter muss diese Weitergabe lizenzrechtlich zugelassen sein.
- **Datenschutz** für Kunden- und Mitarbeiterdaten.
- **Datensicherheit** der IT-Systeme des Auftraggebers und des Auftragnehmers.
- Kompetentes und zertifiziertes Fachpersonal.
- Auch nachvertraglicher, mit Vertragsstrafen bewehrter Know-how-Schutz für Betriebs- und Geschäftsgeheimnisse.
- Regelung zur Überleitung von Mitarbeitern gemäß § 613 a BGB (**Betriebs**-(teil)**übergang**). Festzulegen sind die übergehenden Mitarbeiter, deren Information (§ 613 a Abs. 5 BGB) und eine Regelung zum Übergang der Haftung für Forderungen der übergehenden Mitarbeiter.
- Übergang von Miet-/Leasingverträgen (für Hardware) auf den Auftragnehmer,[259] soweit dieser Systeme übernimmt.

---

[257] *Heymann*/*Lensdorf*, a.a.O., Abschnitt 5.4 Rdn. 215.

[258] Zum Change Management s. S. .., für ITIL ..., für ISO 20000 ....

[259] *Heymann*/*Lensdorf*, Outsourcing-Vertrag, in: Redeker (Hrg.), Handbuch der IT-Verträge, Abschnitt 5.4 Rdn. 268.

- Regelung zur wechselseitigen Information und Abstimmung.[260]
- **Beendigungsunterstützung** durch Auftragnehmer bei Auslaufen des Vertrages/ordentlicher Kündigung/fristloser Kündigung/Wechsel der Plattform; Verpflichtung zur Rückgabe der Daten, Sicherstellen der „Remigrationsfähigkeit", Bereitstellen migrationsfähiger Daten(formate), Mitarbeiterschulung, Rückübernahme der Funktionen („**Backsourcing**", s. unten)[261]; eine Trennung vom Anbieter wird damit selbst zum Projekt. Hier sind in der Praxis am häufigsten Regelungsdefizite zu finden. Hat sich der Auftraggeber im Wege des Betriebs(Teil)übergangs von IT-kompetenten Mitarbeitern getrennt, kann ein Point of no return erreicht sein, ab dem ein Backsourcing nicht mehr möglich ist.[262] Zu regeln ist auch, ob eine Rückübertragung von Software erforderlich sein wird.
- Besondere Regelungen bei grenzüberschreitendem und gar weltweitem Outsourcing (etwa zu Leistungsort, Währung, jeweils anwendbarem Recht und Gerichtsstand).
- Anwendbares Recht.
- Gerichtstand.

i) Typische Fehler in Outsourcing-Projekten

- Unvollständige oder unklare Leistungsbeschreibung,
- unzureichende Beschreibung von Prozessen und Schnittstellen,
- Auslagern ungelöster Probleme,
- zu enger Zeithorizont,
- unzureichend spezifizierte Projektziele und Messkriterien,
- ungenügende Vorbereitung der Mitarbeiter insbesondere der IT-Abteilung,
- zu lange Vertragsbindung,
- ungenügende Größe des Dienstleisters,
- unklare bzw. unvollständige Regelungen zur anbieterseitigen Unterstützung bei Vertragsbeendigung,
- kein klar definiertes Reporting.

j) Backsourcing

Zutreffend wurde bemerkt, dass eine Rücknahme des Outsourcing außerordentlich schwierig ist.[263] Macht ein Unternehmen die Auslagerung von Aufgaben wieder rückgängig, spricht man von „Backsourcing" (bzw. teilweise auch von „Insourcing"). Gründe sind mit Outsourcing verbundene Qualitätseinbußen, höhere Managementkosten durch mehr Koordinations- und Betreuungsbedarf, längere Lieferzeiten und, hierdurch bedingt, kostspielige Lagerhaltung.[264] Beim Outsourcing-Auftraggeber können Kontroll- und Informationsverluste eintreten, und, noch gravierender, Kernkompetenzen in

---

[260] *Heymann*/*Lensdorf*, a.a.O.,Abschnitt 5.4 Rdn. 271f.
[261] *Blöse*/*Pechardscheck*, CR 2002, 785, 790.
[262] Computerwoche 17/2004, 28, 29.
[263] *Stahlknecht*/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 453.
[264] *Etscheit*, Vorreiter des „Insourcing", ZEIT Nr. 20 v. 12.5.2005, 32.

wichtigen Bereichen wie Rechnungswesen oder Informationstechnik abhanden kommen.[265] Im IT-Bereich waren in 2001 noch 86 % der Befragten mit den Ergebnissen eines Outsourcing zufreiden, in 2003 aber nur noch 50%.[266] Diese Schwierigkeiten finden auch zunehmend bei Controllern und Börsenanalysten Beachtung. Nach einigen Untersuchungen soll jedes vierte Unternehmen an Fremdirmen vergebene Aufgaben wieder zurückholen („rezentralisieren").[267]

Ein hauptsächlicher Risikobereich des Backsourcing sind **Transferschwierigkeiten**. So können ehemalige IT-Mitarbeiter verpflichtet sein, wieder zum alten Arbeitgeber zurückzuwechseln (Betriebsrückübergang, der auch § 613 a BGB folgt), aber dem Rückübergang widersprechen. Auch kann der Outsourcing-Kunde verpflichtet sein, seinem bisherigen Anbieter entgangene Gewinne zu ersetzen. Auch muss bei Anbieterwechsel der gesamte Ablauf der Auslagerung erneut durchgeführt werden. Hierzu kann gehören, dass neue IT-Systeme und Softwarelizenzen erworben werden müssen. Schließlich kann eine gewisse Zeit vergehen, bis sich der neue Anbieter erforderliches Fachwissen angeeignet hat.

l) Übergang von Arbeitnehmern

Die Übernahme der Arbeitnehmer ist im Outsourcing-**Vertrag** zu regeln. Hierbei ist zu beachten: Die Aufgabenübertragung auf den Outsourcing-Anbieter kann sich als „Betriebsübergang" im Sinne von § 613a BGB darstellen.[268] Die bisher befassten Mitarbeiter werden hierbei zu Arbeitnehmern des Anbieters. Dies wurde sogar für die Übertragung von (früher selbst wahrgenommenen) Reinigungsaufgaben von einem Unternehmer auf einen anderen durch Vertrag bei Beschäftigung nur eines Arbeitnehmers anwendbar.[269] Der EuGH sieht auch eine reine Funktionsnachfolge als einen entsprechenden Betriebsübergang an. § 613 a BGB ist aber nicht anwendbar, wenn der Auftragnehmer weder Arbeitsmittel noch Personal übernimmt.[270]

Mit Wirkung seit dem 1.4.2002[271] muss der Betriebsveräußerer bzw. der Betriebserwerber bei einem Betriebs(teil)übergang die betroffenen Arbeitnehmer in Textform über die in § 613 a Abs. 5 BGB aufgeführten Punkte unterrichten. Die

---

[265] *Viehöfer*, Selbst ist die Belegschaft, ZEIT Nr. 47 v. 13.11.2003, 22.

[266] Heise online 29.4.2003, Sechs Milliarden Euro durch IT-Outsourcing verschwendet, www.heise.de/newsticker/data/ola-29.04.03-002.

[267] *Viehöfer*, a.a.O.

[268] Grundlage ist die Ratsrichtlinie 77/187/EWG vom 14.2.1977 zur Angleichung von Rechtsvorschriften der Mitgliedstaaten beim Übergang von Unternehmen, Betrieben oder Betriebsteilen.

[269] EuGH, Urt. v. 14.04.1994 – Rs C-392/92, BB 1994, 1500 – Christel Schmidt.

[270] BAG AP BGB § 613 a Nr. 173 = NZA 1998, 536; Dieterich/Hanau/Schaub/Preis, Erfurter Komm. zum Arbeitsrecht, 230 BGB §613 a Rdn. 37.

[271] Gesetz zur Änderung des Seemannsgesetzes und anderer Gesetze v. 23.3.2002 (BGBl 2002 I, 1163) in Umsetzung von Art. 7 Abs. 6 der Richtlinie 2001/23/EG des Rates v. 12.3.2001 zur Angleichung der Rechtsvorschriften der Mitgliedsstaaten über die Wahrung von Ansprüchen der Arbeitnehmer bei Übergang von Unternehmen, Betrieben oder Unternehmens- oder Betriebsteilen, ABl EG Nr. L 82 v. 22.3.2001, 16

Unterrichtung ist Pflicht desArbeitgebers und begründet einen einklagbaren Auskunftsanspruch.[272]

Zugleich haben Arbeitnehmer seit dem 1.4.2002 ein gesetzliches **Widerspruchsrecht**, das freilich bereits von der Rechtsprechung ausgebildet worden war.[273] Bei wirksamer und rechtzeitiger Ausübung des Widerspruchsrechts galt die Zustimmung zum Übergang des Arbeitsverhältnisses auf den neuen Betriebsinhaber als verweigert und bestand das alte Arbeitsverhältnis zum Betriebsveräußerer fort.[274] Das gilt auch nach neuem Recht, wobei allerdings nun die einmonatige Widerspruchsfrist erst ab Zugang der ordnungsmäßigen Unterrichtung läuft (§ 613 a Abs. 6 S. 1 BGB); dies gilt auch dann, wenn die Unterrichtung erst nach Betriebsübergang erfolgt.[275] Eine zeitliche Obergrenze wurde nicht umgesetzt. Ist die Unterichtung also unvollständig oder überhaupt nicht erfolgt, bleibt längerfristig ungewiss, wieviele Arbeitnehmer bei dem Veräußerer bleiben. Angeraten wird, vom Arbeitnehmer für den Zugang eine Empfangsbestätigung unterzeichnen zu lassen.[276] Für den Widerspruch wird Schriftform verlangt (§ 126 BGB), also eigenhändige Unterzeichnung.

3. Agile Softwareentwicklung

3.1 Grundsätze

a) Risiken der Grundprinzipien

Agile Softwareentwicklung soll helfen, den Entwicklungsprozess zu beschleunigen und zugleich flexibler zu gestalten. Im „Manifesto for Agile Software Development“ werden hierfür vier Grundprinzipien formuliert.[277]

aa) Dokumentation
Eher formale Elemente wie vordefinierte Prozesse, Dokumentationen, Vertragsinhalte und Pläne treten in den Hintergrund. Risiko: Die die Vertragsparteien verlieren mangels Pflichtenheft und Dokumentation das

---

[272] *Willemsen*/*Lembke*, NJW 2002, 1159, 1161.

[273] S. etwa BAG NJW 1975, 1378; BAG AP Nr. 103 zu § 613 a BGB, bestätigt durch EuGH NZA 1993, 169.

[274] BAG NJW 1998, 3138f = NZA 1998, 750.

[275] Gesetzesbegr., BT-Drucks. 14/7760, 20.

[276] *Willemsen*/*Lembke*, NJW 2002, 1159, 1160.

[277] Die Grundsätze lauten (*www.agilemanifesto.org*; alle Webseiten-Angaben mit Visit-Datum 20.1.2010):
1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

eigentliche Entwicklungsziel aus den Augen und die Übersicht in Sammlungen von „Nice to have"-Merkmalen.

b) Änderungen

Das Risiko erhöht sich dadurch, dass – ganz im Gegensatz zum „traditionellen" Management der Softwareentwicklung – Änderungen auch in späten Entwicklungsphasen möglich und gewünscht sein sollen.[278] In der Praxis bleibt hier der Aufwand nur dann kalkulierbar, wenn die Änderungen lediglich abgegrenzte kleine Softwaremodule betreffen und nicht oder nur begrenzt mit Rückwirkungen (Side Effects) auf andere Teile des Codes zu rechnen ist. Anderenfalls kann bereits der Aufwand für das ständige Wiederholen erforderlicher Tests und das Umschreiben bereits erstellter Dokumentationsteile exponentiell ansteigen, da sich mit fortschreitendem Entwicklungsstand auch die Möglichkeiten unvorhergesehener Nebenwirkungen durchgeführter Änderungen auf bereits vorhandene Teile des Entwicklungsprodukts entsprechend mehren.

c) Kommunikation

Ein weiteres Prinzip verlangt es, häufig Software an den Kunden auszuliefern[279] und mit diesem täglich(!) in persönlicher Kommunikation[280] zusammenzuarbeiten.[281] Auf den Kunden kommt allein hierdurch ein erheblicher, auch personeller Aufwand zu, der durch die Vorteile des flexiblen Entwickelns zumindest ausgeglichen sein muss. Auch kann er vor dem Problem stehen, die Einigung über bestimmte, erst während der Projektdurchführung ad hoc und nur mündlich geäußerte Leistungswünsche im Streitfall darlegen und beweisen zu müssen.

3.2 Formen

Zu beachten ist, dass es nicht „die", sondern verschiedene agile Entwicklungsmethoden gibt,[282] die einander in wichtigen Merkmalen (etwa dem Verzicht auf das Dokumentieren) teilweise ausschließen, weshalb die jeweils passende gewählt werden muss. Dies zu klären ist primär Kundenrisiko, wobei der Anbieter (in zumutbarem Umfang) hinweis-, aufklärungs- und beratungspflichtig sein kann.

---

[278] Principles behind the Agile Manifesto, Principle 2: Welcome changing requirements, even late in development. Agile Processes harness change for the customer's competitive advantage.

[279] Principle 3.

[280] Principle 6.

[281] Principle 4.

[282] Für eine ausführliche Übersicht über die Merkmale der verschiedenen agilen Entwicklungsmethoden s. *Schneider*, ITRB 2010, 18 (19).

a) eXtreme Programming

Der am weitesten reichende Ansatz ist das von Kent Beck ausgearbeitete eXtreme Programming (XP).[283] Die Anforderungen werden in Form von User Stories möglichst vom Kunden selbst auf Story Cards beschrieben,[284] die allerdings nicht die Dokumentation implementierter Programmteile aufweisen. Die Anforderungen können sich bei Hinzutreten weiterer Benutzerwünsche immer wieder ändern.

XP verlangt ein iteratives und inkrementelles Vorgehen; jede solche Iteration soll etwa zwei bis drei Wochen dauern.[285] Projektplanung erfolgt nicht im Gesamtmodell, sondern von Iteration zu Iteration,[286] ebenso die Aufwandsschätzung, die jeweils vom Kunden gebilligt werden muss.[287] Ausgangspunkt sind in „Quick Design Sessions" festgelegte Entwürfe.[288] Einheitlicher Programmierstil und Konventionen sind einzuhalten. Erforderlich sind Reviews, um Fehler im Code oder Abweichungen von Standards oder in Anforderungen festzustellen. Beim XP kann der Kunde nur den Quellcode und die Unit-Tests als Dokumentationssubstitute erwarten,[289] nicht aber eine Entwicklungsdokumentation – eine für viele Anwendungen unzureichende Basis. Der Kunde tut gut daran, die Erstellung einer Benutzerdokumentation und erst recht, wenn benötigt, einer Entwicklungsdokumentation ausdrücklich zu vereinbaren, die er insbesondere immer dann benötigen wird, wenn er etwa Drittfirmen später mit Softwarepflegeleistungen beauftragen will.

Der Kunde muss außerdem ständig vor Ort sein (On-site Customer) und als Ansprechpartner für fachliche Fragen permanent zur Verfügung stehen, um in den kurzen (u.U. täglichen) Release-Zyklen Rückäußerungen geben zu können. Ein Pflichtenheft existiert nicht. Es wird durch die Kundenantworten ersetzt. Damit kann das Anforderungsprofil der Software (und die Kostenkontrolle) ins Fließen geraten. Die Programmierung wird immer durch zwei Entwickler durchgeführt, von denen der eine codiert, der andere überprüft und analysiert (Pair Programming). Im Zweifel können beide Entwickler Zeugen gegen den Kunden sein, der deshalb u.U. doch den Inhouse-Aufwand treiben muss, seine Äußerungen gegenüber dem Anbieter zu dokumentieren oder akustisch aufzuzeichnen.

Die Entwicklung erfolgt testgetrieben: Programm-Units und Unit-Tests werden stets parallel entwickelt, die Tests also vor Implementieren des jeweiligen

---

[283] Der Ansatz wurde 2004 zu „XP2" weiterentwickelt.

[284] *Bleek*/*Wolf*, Agile Softwareentwicklung 2008, 142.

[285] *Hruschka*/*Rupp*/*Starke*, Agility kompakt, 2. Aufl. 2009, 85.

[286] *Hruschka*/*Rupp*/*Starke*, Agility kompakt, 2. Aufl. 2009, 87.

[287] *Hindel*/*Hörmann*/*Müller*/*Schmied*, Geplante Selbstorganisation, in: iX Kompakt IT-Projekte 3/2009, 80 (83).

[288] *Bleek*/*Wolf*, Agile Softwareentwicklung 2008, 86.

[289] *Balzert*,*www.w3l.de/W3lmedia/W3L/Medium164417/folien*;Wikipedia, *http://de.wikipedia.org/wiki/Testgetriebene Entwicklung*.

Moduls;[290] aus zu erarbeitenden Testfällen wird der Code erstellt.[291] Die Entwicklung erfolgt in „Mikro-Iterationen". Erzeugt wird eine „ausführbare Spezifikation". Regelmäßig werden Wiederholungen von Programmteilen entfernt und sonstige Optimierungen durchgeführt (Refactoring), die freilich als solche zu Fehlern führen können, was durch automatisiert ablaufende Unit-Tests vermieden werden soll. Bei testgetriebener Entwicklung soll die Möglichkeit bestehen, eine in die Entwicklung integrierte Qualitätssicherung zu generieren,[292] jedoch bedarf dies grundsätzlich besonderer vertraglicher Vereinbarung. Auch bleibt zu prüfen, ob die allgemeinen Anforderungen nach DIN/ISO 9001 und 20000 erfüllt werden (können).

b) Industrial XP

Mehr Kontrolle des Projektverlaufs ermöglicht die XP-Variante Industrial XP (IXP). Sie nimmt Praktiken aus den anderen (nachfolgend erwähnten) Verfahren auf, nämlich Projektskizze (als einfaches Pflichtenheft), Risikomanagement, testgetriebenes Entwickeln, fachlich getriebener Entwurf, Identifikation der Entwickler mit Qualitätsarbeit.[293] Dadurch wird das Entwickeln langsamer, aber deutlich besser kontrollierbar.

c) Feature Driven Development

Das Feature Driven Development strukturiert den Entwicklungsprozess stärker[294] und stellt primär auf Festpreiskonstellationen mit fixiertem Umfang ab.[295] Zuerst wird mit dem Kunden ein Gesamtmodell der gewünschten Anwendung entwickelt und auf dessen Basis eine Feature-Liste planend, entwerfend und entwickelnd abgearbeitet.[296] Die Entwicklung ist durch ihre übersichtliche Granularität des iterativen und inkrementellen Entwickelns[297] leichter kontrollierbar (und dokumentierbar), jedoch muss der Kunde spätestens mit Abschluss des Gesamtmodells genau wissen, was er will.[298] Außerdem ist an keiner Stelle im Entwicklungsprozess ein vollständiger Entwurf vorgesehen, wodurch das Erreichen eines einheitlichen Designs erschwert wird und sich Features nach

---

[290] Wird eine Entwicklungsumgebung (Framework) wie JUnit eingesetzt, müssen die Tests möglichst schnell lauffähig gemacht werden, sodass im Test-Framework „der Balken grün" wird (*Hruschka*/*Rupp*/*Starke*, Agility kompakt, 2. Aufl. 2009, 89). Richtig verstanden zeigt dieser Hinweis, dass die Testfälle eigentlich kleine Anwendungsteile (Units) sind, nicht reine Prüftests aus Kundensicht, ob fertig erstellter Code funktioniert. Vielmehr wird solange Code geändert, bis der Test „läuft". Der Kunde muss hier prüfen können, ob das Programm an die Anforderungen angepasst wird und nicht umgekehrt die Anforderungen an die lauffähigen Testteile.

[291] *Hruschka*/*Rupp*/*Starke*, Agility kompakt, 2. Aufl. 2009, 86.

[292] S. die Hinweise bei *http://www.iur-it.de/beitraege/Agile_Total_Quality*.

[293] *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl 2008, 660f.

[294] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 90.

[295] *Bleek*/*Wolf*, a.a.O., 152.

[296] *Bleek*/*Wolf*, a.a.O., 153.

[297] *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement 2. Aufl 2008, 667.

[298] *Bleek*/*Wolf*, Agile Softwareentwicklung 2008, 153.

Fertigstellen als unsinnig herausstellen oder Inkonsistenzen auftreten können.[299] Restrukturieren ist nur begrenzt möglich. Anstelle des paarweisen Programmierens werden formale Inspektionen durchgeführt;[300] hier kommt der Kunde einer Abnahme recht nahe, sofern er ständig in den Entwicklungsprozess einbezogen wird. Freilich wirkt eine solche Billigung als im Wesentlichen vertragsgerecht nur für den jeweils kundenseits prüfbaren Programmteil, weshalb der Kunde selbst festhalten sollte, wann er welche Teile (mit-) geprüft hat. Der Kunde legt die Prioritäten bei den Geschäftsprozessen fest, der entwickelnde Anbieter die Prioritäten der Features.

3.3 Methoden

a) Scrum

Das oft erwähnte „Scrum“ (Gedränge, abgeleitet aus dem Rugby) stellt keine spezifische softwarebezogene Entwicklungsmethode dar, sondern einen Managementrahmen.[301] Dieser Ansatz erhöht die Übersichtlichkeit, da alle Anforderungen in einem „Product Backlog“ (anstatt in User Stories) gesammelt werden (gewissermaßen ein „lebendes“ Pflichtenheft). Aus dem Product Backlog werden die im nächsten Entwicklungsschritt (Sprint) abzuarbeitenden Anforderungen im „Sprint Backlog“ gesammelt, im Sprint als Umsetzung je einer Iteration bearbeitet und im Sprint-Review überprüft.[302] Täglich sollen Gespräche (hauptsächlich zwischen Entwicklern, nicht notwendig mit dem Kunden) stattfinden (Daily Scrum). Jede Sprint-Phase ist rückblickend zu beurteilen (Retrospektive).

b) Crystal Family

Für die agile Softwareentwicklung eingesetzt werden können verschiedene Verfahren. Ein Beispiel ist die nach der Anzahl der beteiligten Personen differenzierende „Crystal-Family“.[303] Anders als etwa bei XP ist bei Crystal ein Entwickeln beim „On-site Customer“ und das „Pair Programming“ nicht erforderlich. Viel verwendet wird auch das in der Programmiersprache „Ruby“ geschriebene, quelloffene Web Application Framework „Ruby on Rails“ (oder kurz „Rails“),[304] das mit einer Art Gerüstbau (Scaffolding) das Schreiben von

---

[299] *Obendorf*, Szenariotechniken & Agile Softwareentwicklung, in: Gross (Hrg.), Mensch & Computer 2007, Konf. für interaktive und kooperative Medien 2007, 19, *http://mc.informatik.uni-hamburg.de/konferenzbaende/ mc2007/konferenzband/mc2007_04_Obendorf.pdf*, der von einer „Scheu“ vor einem „big upfront Design“ spricht. Hierdurch kann sich freilich für den Kunden wirtschaftlich der Korrektur- und Anpassungsaufwand erhöhen.

[300] *Balzert*, *www.w3l.de/W3lmedia/W3L/Medium164417/folien*.

[301] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 149.

[302] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 151.

[303] Für alle *Cockburn* in: *http://alistair.cockburn.us/Crystal+Clear+distilled*; *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl 2008, 674.

[304] Für eine Übersicht s. *http://de.wikipedia.org/wiki/Ruby_on_Rails*. Twitter wurde etwa mit „Rails“ entwickelt.

Web-Applikationen „on the Fly“ erlaubt und eine sofortige Visualisierung ermöglicht, aber nur eine Inline-Dokumentation des Quellcodes.[305]

3.4 Auswirkung agiler Entwicklungsverfahren auf die Steuerung des IT-Projekts

Für jedes der nachfolgenden Projektmerkmale muss der Kunde projektspezifisch prüfen, ob er daruf verzichten kann bzw. ob die Anforderungen ausreichend konkret geregelt sind.

a) Kein Lasten- und IT-Pflichtenheft

Ein gewisser verborgener Widerspruch in der Sicht des Anbieters auf den Kunden scheint symptomatisch. Einerseits soll der Kunde bzw. der zuständige Mitarbeiter fachlich kompetent, ständig präsent und kooperationsbereit sein. Andererseits wird das flexibel-agile Entwickeln wohl meist nur dann erforderlich, wenn der Kunde noch keine klaren Vorstellungen von der gewünschten Anwendung hat. Eine Bemerkung aus der Praxis macht dies schlaglichtartig deutlich: „Mit Version 1.0 bekommt der Kunde eigentlich nicht, was er will, sondern das, was im Vertrag steht“.[306] Genau dieses Verwendungsrisiko (etwa anderes vereinbart zu haben als er braucht) ist freilich typischerweise dem Kunden zugeordnet.

Gegenüber einem solchen Kunden bleibt beim klassischen Vorgehen nur, Schritt für Schritt herauszufinden, was der Kunde tatsächlich benötigt, und sukzessive entsprechenden Change Requests abzuarbeiten. Zu Beginn dieses Prozesses wissen beide Seiten noch nicht, wohin sie die Klärung führen wird. Dieser Klärungsprozess nach Vertragsschluss und erfolgter Erstellung des Produkts wird nun bei Einsatz agiler Erstellungsmethoden in den Erstellungsprozess selbst vorverlagert. Die Unsicherheit über das Entwicklungsziel bleibt aber die gleiche. Bei herkömmlicher Programmerstellung auf der Basis von Lastenheft und Spezifikation muss der Kunde Änderungsaufträge erteilen und diese Änderungen voll vergüten. Ähnlich trägt der Kunde grundsätzlich auch bei agiler Entwicklung das Risiko der hieraus entstehenden Kosten für das nachträgliche Einfügen von Features. Der Anbieter kann insoweit nicht das Erreichen eines bestimmten Leistungsziels versprechen (und auch keinen Leistungserfolg schulden), da dieses erst gemeinsam im laufenden Vertrag festgelegt wird. Der Kunde kann seinerseits nicht vorab feststellen, mit wievielen kostenauslösenden Iterationen er rechnen muss. Außerdem erschwert eine Entwicklung „Feature by Feature“ ein einheitliches Design.[307]

---

[305] *Henning/Vogelsang* (Hrg.), Taschenbuch Programmiersprachen, 2. Aufl. 2007, 507.

[306] *www.ordix.de/ORDIXNews/3_2007/Projektmanagement/agile-software-entwicklung.html.*

[307] *Obendorf*, Szenariotechniken & Agile Softwareentwicklung, in: Gross (Hrg.), Mensch & Computer 2007, Konf. für interaktive und kooperative Medien 2007, 19, *http://mc.informatik.uni-hamburg.de/konferenzbaende/ mc2007/konferenzband/mc2007_04_Obendorf.pdf.*

Der Verzicht auf Pflichtenhefte bedeutet freilich nicht, dass Entwickler einfach ins Blaue hinein codieren sollen bzw. dürfen. Vielmehr wird eine gemeinsame Grundvorstellung von der gewünschten Anwendung bzw. den Geschäftszielen vorausgesetzt, deren Bezeichnung mit unscharfen Begriffen als „Vision“ oder „Metapher“ (in XP) freilich für manche gewöhnungsbedürftig ist. Diese Zielvorgabe sollte soweit konkretisiert werden, dass sich aus den Geschäftszielen Anforderungen formulieren lassen und in etwa die Anzahl voraussichtlicher Iterationen absehbar und für beide Vertragspartner kalkulierbar wird. Bisher war dieses Ergebnis als „fachliche Feinspezifikation“ bekannt.

Auch mit dieser Konkretisierung verbleiben Risiken. Teilt man ein Projekt in viele Einzelschritte ohne übergeordnetes Gesamtdesign auf, entsteht leicht das Risiko, sich im Kreis zu bewegen und dem benötigten Programm zu langsam näher zu kommen.[308]

b) Lauffähige Software statt Dokumentation

IT-Vertragsrechtler sind quasi von Anfang an mit der fundamentalen Bedeutung der Dokumentation sozialisiert worden. Entsprechend schwer wird es ihnen fallen, pauschal das Dokumentieren als verzichtbar behandeln zu müssen.

Bei Überlassung einer Standardsoftware ist der Anbieter grundsätzlich zur Lieferung einer Bedienungsanleitung (bzw. eines Benutzerhandbuchs) verpflichtet, anderenfalls erfüllt er den Vertrag (jedenfalls teilweise) nicht.[309] Eine Entwicklungsdokumentation ist zumindest dann geschuldet, wenn der Anbieter nicht die Pflege der (aktualisierungsbedürftigen) Software übernimmt und der Kunde alle Rechte an der Software eingeräumt erhalten hat. [310] Dies wirft die Frage auf, ob die Wahl agiler Methoden der Softwareentwicklung den Anbieter ohne Weiteres von dieser vertraglichen Dokumentationspflicht befreien kann. Hier ist zu differenzieren:

Allein daraus, dass der Anbieter ein agiles Entwicklungsverfahren anbietet, muss für den Kunden noch nicht erkennbar sein, dass er nicht mit einer Benutzerdokumentation rechnen darf (wenn er nicht ausnahmsweise selbst im Bereich der Software-Entwicklung tätig ist). Selbst das Agile Manifesto schließt eine Dokumentation nicht generell aus, sondern behandelt sie in der Gewichtung nur nachrangig.

Der Anbieter muss deshalb dem Kunden ausdrücklich zu erkennen geben, dass die Dokumentationserstellung nicht zur gewählten Leistungsform gehören soll. Dies

---

[308] Anschaulich macht dies *Ebert*, Systematisches Requirement Management 2005, 245f: Viele Prozesse wachsen über die Zeit stark an, weil man sich nie fragte, was davon inzwischen unnötig ist. Auf jeden Fehler im Produkt kam ein weiterer Testschritt. Mit jeder falsch verstandenen Anforderung aus einem Kundenkontakt hat sich die Checkliste vergrößert. Mit jedem größeren Projekt haben sich Templates und Entwicklungsdokumentationen vergrößert. Verkleinert und vereinfacht werden sie nie automatisch.

[309] BGH, Urt.v. 4.11.1992 – VIII ZR 165/91, CR 1993, 203.

[310] *Redeker*, IT-Recht, 4. Aufl. 2007, Rdnr. 315; *Koch*, Computer-Vertragsrecht, 7. Aufl. 2009, S. 78.

kann durch einen ausreichend konkreten Hinweis und eine Aufklärung über die Auswirkungen des Fehlens einer Dokumentation in den Vertragsverhandlungen geschehen; hierfür wird der Anbieter freilich beweispflichtig sein.

Denkbar ist die vertragliche Vereinbarung eines Verzichts auf die Dokumentation.[311] Hierbei sollte ausdrücklich geregelt sein, ob sich dieser Verzicht auf jede Art von Dokumentation bezieht oder etwa nur auf die Entwicklungsdokumentation. Selbst dann kann eine solche Regelung problematisch sein, wenn sie im Formularvertrag erfolgen soll. Hierin kann nämlich eine unangemessene Benachteiligung des Kunden i.S.v. § 307 Abs. 2 Nr. 2 BGB durch die Einschränkung eines wesentlichen Rechts, nämlich eines (zumindest teilweisen) Nichterfüllungsanspruchs zu sehen sein. Der Anbieter wird deshalb eine individualvertragliche Regelung vorziehen.[312] Hier läuft er freilich das Risiko, den Verzicht nicht umfassend genug oder zu intransparent zu formulieren, sodass er nicht beweisbar ist.

Als Alternative bleibt, anbieterseits die Leistungsbeschreibung so zu fassen, dass ein rasches und nicht durch Verwaltungsaufwand für Dokumentieren belastetes Entwickeln angeboten wird. Auch dies ist nicht ohne Risiko, wenn oder soweit dieses Vorgehen als Umgehen des Verbots aus § 307 Abs. 2 Nr. 2 BGB gesehen werden kann.

Und schließlich verlangen agile Methoden keinen vollständigen Verzicht auf das Dokumentieren. Selbst energische XP-Anhänger betrachten etwa bei Entwicklung und Test einer Komponente die Dokumentation als „selbstverständlich“ dazu gehörend.[313] Ein vollständiger Ausschluss einer Dokumentationspflicht würde deshalb das Übermaßverbot verletzen.[314] Wenn der Verzicht aber nur für bestimmte Dokumentationsteile gelten soll, bleibt es grundsätzlich das Risiko des Anbieters, diese Abgrenzung klar erkennbar zu ziehen und den Kunden über die Auswirkungen dieses partiellen Verzichts aufzuklären. Der Anbieter trägt damit das rechtliche Risiko, sich nicht voll von seiner Dokumentationspflicht entlasten zu können.

c) Keine Qualitätssicherung

Anbieter sind bei der Softwareentwicklung zur Qualitätssicherung verpflichtet,[315] auch sie ist mithin nicht umstandslos verzichtbar.

---

[311] So auch *Schneider*, ITRB 2010, 18, 21.

[312] S. näher *Schneider*, ITRB 2010, 18, 21.

[313] *Hruschka*/*Rupp*/*Starke*, Agility kompakt, 2. Aufl. 2009, 90.

[314] *Fuchs* in: Ulmer/Brandner/Hensen, AGB-Recht, 10. Aufl. 2006 § 307 Rdnr. 106.

[315] *Schneider*, Handbuch des EDV-Rechts, 4. Aufl. 2009, E Rdnr. 339 a; *Koch*, Computer-Vertragsrecht, 7. Aufl. 2009, S. 384.

Auffallend ist schon, dass in den Darstellungen zur agilen Softwareentwicklung das Thema Qualitätssicherung kaum eine Rolle spielt. Agilität scheint sich sogar umgekehrt poportional zur Qualität zu verhalten: Mit steigender Agilität sinkt die Qualität. Gerade Stimmen aus der Praxis belegen dies deutlich: „Man will schnelle Ergebnisse haben. Dafür opfert man Qualität der Software." „Durch schnelles Entwickeln geht man das Risiko ein, nicht alles ausgereift implementieren zu können."[316] XP legt Wert auf den möglichst einfachen Entwurf, nicht auf die Qualität der Erstellungsprodukte. Fraglich ist aber, ob diese scheinbar etwas laxen Ansätze den Anbieter von vertraglicher Erfüllungs- oder gar Produkthaftung entlasten können. Dies wird grundsätzlich wohl zu verneinen sein, wenn die Vertragsparteien nicht ausdrücklich etwas anderes vereinbart haben. Da die Vereinbarung Kundenrechte wesentlich einschränken kann, wird sie, wie ausgeführt, nur individualvertraglich wirksam getroffen werden können. Gegenüber geschädigten Dritten kann sich der Anbieter ohnehin nicht freizeichnen.

Damit haftet der Anbieter grundsätzlich auch bei Einsatz agiler Softwareentwicklungsmethoden aus Fehlern dieser Software und allgemein für die Organisation und Überwachung auch der Qualitätssicherung. Insbesondere gegenüber geschädigten Dritten kann es nicht entlastend wirken, wenn der Anbieter besonders schnelle, damit aber in erhöhtem Maße fehlergeneigte Verfahren anwendet.[317] Ihn trifft die Beweislast dafür, dass der Fehler nicht in seinem Verantwortungsbereich entstanden ist.[318] Auch gehört das Rückverfolgbarmachen der einzelnen Entwicklungsschritte zwecks Fehlerverfolgung zur Anbieterpflicht, um etwa sicherzustellen, dass Software gar nicht erst an Kunden ausgeliefert wird, für die andere Kunden bereits (gar sicherheitsgefährdende) Mängel gerügt haben.

d) Intensive und ständige Kundenmitwirkung

Kunden machen sich oft nicht klar, dass ihre Mitwirkung in agilen Entwicklungsprojekten deutlich intensiviert ist.[319] Sie müssen möglichst ständig als Ansprechpartner vor Ort sein[320] – das bindet Mitarbeiter. Sie müssen außerdem angesichts häufiger Entwicklungszyklen laufend Programmfassungen überprüfen und akzeptieren oder ablehnen und hierzu entscheidungsberechtigt sein. Schließen sollten die beteiligten Mitarbeiter des Kunden Fachexperten der gewünschten Anwendung sein.[321] In manchen Fällen kann zu fragen sein, ob dieser Aufwand

---

[316] *Murawski*, www.inf.fu-berlin.de/inst/ag-sc/teaching/S-AGILE/-2004/ausarbeitungen/Stefan_Murawski_agile_prozesse_ausarbeitungen_final-pdf.

[317] Dieses Risiko trägt auch der Kunde, der agil entwickelte Software in seinen Produkten einsetzt (etwa in Fahrzeug-Bremssystemen).

[318] So etwa BGH, Urt.v.9.5.1995 – VI ZR 5/91 – Mehrweg-Mineralwasserflaschen, ZIP 1995, 1094, 1097.

[319] Zur Kundenmitwirkung und möglichen Eskalationsregelungen s. ausf. *Lapp*, ITRB 2010, 69 (70).

[320] Denn ein „befragbarer" Kunde soll Teile der Spezifikation ersetzen, *www.ordix.de/ORDIXNews/3_2007/Projektmanagement/agile-software-entwicklung.html*.

[321] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 148.

nicht besser in eine Lastenhefterarbeitung und Anforderungsspezifikation vor Beginn der Software-Erstellung investiert werden sollte. Dann muss der Kunde auch weniger mit Problemen durch seine innerbetriebliche Revision und das Controlling rechnen, die bei Programmentwicklung ohne konkrete und in den Leistungsmerkmalen überprüfbare Zielvorgaben schnell in Schwierigkeiten geraten, weil sie bei solcherart "unsicheren" Entwicklungsprozessen mit deutlich erhöhten Risiken hinsichtlich Kostenexplosion oder Scheiterns des Projekts rechnen müssen.

Vertragsrechtlich kann die intensivierte Kooperation die Mitwirkung des Kunden von einer sonst meist gegebenen Obliegenheit zu einer Mitwirkungspflicht aufwerten. Hierfür reicht aber nicht aus, dass der Kunde ein Interesse an ausreichender Leistungsspezifikation haben muss, sondern es muss fallspezifisch ein besonderes Interesse des Anbieters an der Projektdurchführung (etwa bei einem markterschließenden Pilotprojet) hinzukommen.[322]

3.5 Vertragsrechtliche Gestaltung agiler Softwareentwicklung

a) § 651 BGB

Witte hat den Bezug agiler Softwareentwicklung zu § 651 BGB näher dargestellt.[323] Er zeigt zugleich das Dilemma auf, dass sich eine auf Werkvertragsrecht abzielende, immer genauere Leistungsspezifikation und Pflichtenkonkretisierung immer weiter vom flexiblen Ansatz agiler Entwicklung entfernt.

Im Einzelfall zu prüfen ist freilich, ob die etwa bei XP erforderliche Entwicklungsarbeit vor Ort beim und mit dem Kunden auf dessen Systemen überhaupt mit einer „Lieferung“ i.S.v. § 651 Satz 1 BGB verbunden ist. Geliefert wird hier allenfalls eine Basisfunktionalität, die dann mit dem Kunden iterativ angepasst wird. Eine solche Änderung des beim Kunden installierten „Vorprodukts“ fällt nicht mehr unter § 651 BGB.[324] Auch das laufende Sammeln von Testfällen und entsprechende Dokumentieren kann als solche Änderung anzusehen sein. Auch greift § 651 BGB nicht ein, wenn die Mitarbeiter des Anbieters zwar nicht vor Ort beim Kunden tätig werden, aber ihm ihre (u.U. täglichen) Releases online zur Prüfung und Zustimmung übertragen, da dies keine „Lieferung“ i.S.v. § 651 BGB darstellt.[325]

---

[322] Ähnlich i.E. *Redeker*, IT-Recht, 4. Aufl. 2007, Rdn. 417.

[323] *Witte*, ITRB 2010, 44.

[324] *Koch*, Computer-Vertragsrecht, 7. Aufl. 2009, S. 475.

[325] *Koch*, Computer-Vertragsrecht, 7. Aufl. 2009, S. 472f. *Schweinoch*, CR 2010, 1, hält eine „Lieferung“ auch online für möglich; dies ergebe sich aus BGH, Urt.v.18.10.1989 – VIII ZR 325/88, CR 1990, 24. Diese Entscheidung betrifft aber nur die (zumal entsprechende) Anwendung von Kaufrecht (vom AbzG), nicht § 651. BGH, Urt. v. 23.7.2009 – VII ZR 151/08, CR 2009, 637, bezieht sich andererseits nur auf bewegliche Sachen. Aus der BGH-Rechtsprechung ist damit nicht ableitbar, dass § 651 BGB analog auf die Online-Übertragung von Computerprogrammen anwendbar wäre. § 651 BGB beruht auf der Verbrauchsgüterkaufgewährleistungs-Richtlinie 1999/44/EG. Der dort in Art. 1 Abs. 2b als „bewegliche

b) Anwendbares Vertragsrecht

Nach den allgemeinen Grundsätzen schuldet der Anbieter bei Fehlen des Pflichtenhefts aus Werkvertrag nur einen mittleren Ausführungsstandard.[326] Bei agiler Programmentwicklung hilft der Rückgriff auf einen solchen Standard aber allenfalls in ungeklärt gebliebenen Teilbereichen der Entwicklung, da die Ausgestaltung der Software gerade in Abstimmung mit dem Kunden anstelle der Pfichtenhefterstellung erfolgen soll. Gewissermaßen erstellen Anbieter und Kunde die benötigte Zusammenstellung der Leistungsmerkmale gemeinsam.

Einiges spricht dafür, dass zwar das Entwickeln einer ersten Fassung einer lauffähigen und demonstrierbaren Grundfunktionalität Werkvertragsrecht (oder über § 651 BGB Kaufrecht) folgt, die – zumal mündlich gesteuerte, quasi auf Zuruf erfolgende – Weiterentwicklung und Ausgestaltung der ersten Fassung hingegen Dienstvertragsrecht.[327] Die Leistungen sind freilich auch dann unter Einhalten der Maßstäbe des Stands der Technik der Softwareerstellung zu erbringen; die Nichteinhaltung dieser Maßstäbe wird regelmäßig eine anbieterseitige Pflichtverletzung darstellen. Jedoch bedürfen wohl Dokumentierung und Qualitätssicherung unter Dienstvertragsrecht in der Regel gesonderter Vereinbarung (und Weisung).

Bei besonders intensiver Zusammenarbeit von Anbieter und Kunden wurde auch die Annahme einer beide Seiten umschließenden BGB-Gesellschaft als möglich angesehen.[328]

3.6 Indikationen und Kontraindikationen

Betrachtet man die Praxis der agilen Softwareentwicklung näher, so lassen sich einige Kriterien für Fälle formulieren, in denen agile Verfahren nicht gewählt werden sollten:

- Zwingend dokumentierte Erfüllung bestimmter IT-Prozessabläufe und Vorgaben für die Qualität und insbesondere für die IT-Sicherheit (oder gar lebenskritische Funktionen etwa in der Medizin oder Flugüberwachung).[329]

---

körperliche Gegenstände“ definierte Begriff der „Verbrauchsgüter“ kann nicht durch ein deutsches Gericht analog auf unkörperliche Online-Übertragungen ausgeweitet werden.

[326] S. näher *Schneider*, ITRB 2010, 18, 19; *Auer-Reinsdorff* , ITRB 2010 für Entwicklung unter Scrum.

[327] Womit ab diesem Zeitpunkt der Weiterentwicklung die Verweisung des § 651 BGB ohnehin an Bedeutung verliert. Auch wenn die Ausgestaltung Werkvertragsrecht folgt, gelangt man nicht zu § 651 BGB, soweit keine Lieferung erfolgt, sondern die Anpassungen on site beim Kunden durchgeführt werden.

[328] *Lapp*, ITRB 2010, 69 (70), auf gemeinsame Rechte am Code abstellend. Allerdings verbleibt die Kundenmitwirkung meist nur im Bereich der (sukzessiven) Festlegung von Anforderungen, während allein die Mitarbeiter des Anbieters Programmcodes erstellen (oder im objektorientierten Bereich Klassen und Vererbungshierarchien spezifizieren).

[329] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 160.

- Budgetmäßig, terminlich, qualitätsbezogen und vom Umfang her klar fixierte Projekte.[330]
- Langwierige Change Request-Verfahren und lange oder formalisierte Entscheidungswege beim Kunden.[331]
- Kein oder nur unregelmäßiger Kontakt zum Kunden[332] oder Kunden, die an Lastenhefterstellung gewöhnt sind.[333]
- Keine Möglichkeit der Arbeit vor Ort beim Kunden.[334]
- Lange Entwicklungs(Build)-Zeiten.[335]

In diesen Fällen können aber dennoch flexible Entwicklungsmethoden eingesetzt werden, etwa im Modell des Rational Unified Process (RUP). Der RUP gliedert auch in Phasen und diese in Iterationen auf, die definierte und überprüfbare Meilensteine darstellen und jeweils zu „Artefakten" (Produktteilen) als Modellelemente von CASE-Werkzeugen führen.[336] RUP baut wesentlich auch auf den Elementen Anforderungsmanagement, iterative Entwicklung, Qualitätskontrolle, Change- bzw. Konfigurationsmanagement und einer Sammlung von Best Practices auf[337] – und könnte deshalb für viele Kunden das Entwicklungsverfahren der Wahl sein.

Empfohlen wird die agile Entwicklungsmethodik in folgenden Fällen:
- Möglichst kurzfristige Exploration und Abschluss eines Projekts mit einem örtlich konzentrierten kleinen Team unter Kundenbeteiligung.[338]
- Komplexe Produktions- oder Entwicklungsprozesse, die sich weder insgesamt noch in Einzelschritten im Voraus planen lassen.
- Vorliegen eher weicher und wenig ausformulierter Anforderungen, die sich während der Erstellung ändern.[339]
- Entbehrlichkeit des Pflichtenhefts (keine Ausschreibung).

---

[330] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 163.
[331] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 161.
[332] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 161.
[333] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 162.
[334] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 163.
[335] *Bleek*/*Wolf*, Agile Softwareentwicklung, 2008, 165.
[336] *Hindel*/*Hörmann*/*Müller*/*Schmied*, Geplante Selbstorganisation, in: iX Kompakt IT-Projekte 3/2009, 80 (82).
[337] *Hindel*/*Hörmann*/*Müller*/*Schmied*, Geplante Selbstorganisation, in: iX Kompakt IT-Projekte 3/2009, 80 (83).
[338] *Ebert*, Systematisches Requirement Management 2005, 249.
[339] *Balzert*, Lehrbuch der Softwaretechnik. Softwaremanagement, 2. Aufl 2008, 683.

Teil IV

Rechtliche Verantwortlichkeit der Geschäftsleitung für die Projektdurchführung

Gute IT-Projektverträge können für das auftraggebende Unternehmen überlebenswichtig sein. Werden erforderliche vertragliche Regelungen etwa zur zeitnahmen Projektkontrolle oder Maßnahmen zur technischen Angriffssicherheit des betrieblichen IT-Systems nicht betroffen oder Mängelrechte nicht konsequent verfolgt, kann dies die persönliche Haftung der Mitglieder der Geschäftsführung gegenüber der Gesellschaft begründen. Die Mitglieder der Geschäftsleitung des auftraggebenden Unternehmens müssen damit ein eigenes, persönliches Interesse daran haben, ihrer möglichen haftungsweisen Inanspruchnahme bereits in der Phase der Vertragsverhandlungen mit Anbietern durch richtige Vertragsgestaltung und bei der Projektdurchführung durch richtige Kontrolle der Vertragserfüllung vorzubeugen.

Die Steuerung von Risiken aus IT-Projekten ist genuine Aufgabe der Geschäftsleitung, also etwa der Geschäftsführer der GmbH oder des Vorstandes der AG (§ 76 f AktG). Diese Pflicht ergibt sich gegenüber Arbeitnehmern aus dem Arbeitsverhältnis und den hierdurch begründeten Arbeitgeberpflichten, gegenüber Kunden oder sonstigen Dritten aus Vertrag oder Deliktsrecht. Der Geschäftsführer muss auch hierbei die Sorgfalt eines ordentlichen Kaufmannnes walten lassen (§ 43 Abs. 1 GmbHG).

Vorstände von Aktiengesellschaften bzw. GmbH-Geschäftsführer haben bei ihrer Geschäftsführung die Sorgfalt eines ordentlichen und gewissenhaften Geschäftsleiters anzuwenden (§ 93 Abs. 1 S. 1 AktG). Das Erfordernis eine **informierten Entscheidung** verlangt, dass die Geschäftsleitung die erforderlichen Informationen beschafft und Risiken für ein präventives Handeln rechtzeitig prüft. Ein Verzicht auf die Erfüllung dieser Pflichten – etwa durch Nichteinrichten eines Risikofrühwarnsystems gemäß § 91 Abs. 2 AktG – würde die Sorgfaltspflicht des Geschäftsleiters verletzen. Die Geschäftsleitung hat ihre Maßnahmen – schon aus Gründen der Möglichkeit einer Beweisführung – aussagefähig zu dokumentieren.

Weiter darf die Geschäftsleitung **nicht** ohne angemessenen Grund **auf Rechtspositionen** und Ansprüche des Unternehmens **verzichten** (Rechtsdurchsetzungspflicht).[340] Hierzu gehört etwa, dass begründete Ansprüche aus Nicht- oder Schlechterfüllung von IT-Projektverträgen gegen Anbieter in geeigneter Weise durchgesetzt werden müssen (um Schaden vom Gesellschaftsvermögen zu wenden). Zur allgemeinen Sorgfaltspflicht der Geschäftsleitung gehört andererseits, rechtzeitig noch bei den Verhandlungen über den abzuschließenden Projektvertrag für eine ausreichende vertragsrechtliche Absicherung des Auftraggebers zu sorgen, etwa

[340] *Rodewald*, BB 2006, 113, 114.

durch detaillierte und kontrollfähige Leistungsbeschreibung, Definieren eines kompletten Projektzeitplans, genaue Zuordnung von Verantwortlichkeiten etwa für die Pflichtenhefterstellung, Festlegen aussagefähiger Abnahme-/Funktionsprüfungskriterien, Absicherung gegen Leistungs- bzw. Nacherfüllungs-/Mängelbeseitigungsverzug etwa durch Vertragsstrafen, Vereinbaren eines „Ausstiegs"verfahrens bei Nichterreichen der Projektziele.

Bei **Verletzung** dieser Pflicht durch Nichtanwendung der erforderlichen Sorgfalt haften die Mitglieder der Geschäftsleitung gegenüber dem Unternehmen, für das sie handeln, persönlich auf Ersatz entstandener Schäden (§§ 93 Abs. 1 und 2 AktG für Vorstände, ähnlich § 43 Abs. 1 und 2 GmbHG für Geschäftsführer). Voraussetzung der persönlichen Haftung von AG-Vorständen oder GmbH-Geschäftsführern ist freilich, dass durch die Pflichtverletzung unternehmensgefährdende Risiken für das Unternehmen entstehen und diese Pflichtverletzung von dem oder den Mitglied oder Mitgliedern der Geschäftsleitung zu vertreten ist. Dieser Fall kann eintreten, wenn das Scheitern eines nicht ausreichend kontrollierten Projekts den Produktionsbetrieb des Unternehmens gefährdet.

Der **Aufsichtsrat** einer AG hat gemäß § 111 Abs. 1 AktG kontrolliert die Geschäftsführung und ordnungsgemäße Pflichterfüllung des Vorstandes zu kontrollieren (§§ 76, 77 AktG). Der Vorstand muss den Aufsichtsrat über zu treffende Maßnahmen und grundsätzliche Fragen unterrichten (§ 90 Abs. 1 Nr. 1 AktG). Hierzu gehört die Prüfung, ob der Vorstand seinen Pflichten nachkommt und insbesondere gemäß § 91 Abs. 2 AktG ein **Risikofrüherkennungssystem eingerichtet**[341] und sinnvoll organisiert hat.[342]

Die Durchführung von IT-Projekten ist geradezu ein Musterbeispiel für Risiken, die jede Geschäftsleitung früherkennen und begrenzen muss. Diese Projektrisiken wiegen umso schwerer, je stärker ein Scheitern des jeweiligen Projekts den Bestand den auftraggebenden Unternehmens gefährden kann (etwa, weil eine notwendige Migration auf andere Systemplattformen oder die benötigte Einführung oder Änderung von unternehmenssteuernder Software scheitert). Die Durchführung von IT-Projekten muss deshalb von eine Risikomanagement begleitet werden; notwendig ist eine Zuordnung der IT-Risikoszenarien zu Projektthemen.[343]

Zu **Risiken** von IT-Projekten gehört etwa die Entscheidung für eine Technologie, die sich am Markt nicht durchsetzen kann (und deshalb z.B. nicht langfristig unterstützt wird).[344] Ein anderes Risiko kann sich aus der zeitlich parallelen, aber unabgestimmt nebeneinander laufenden Durchführung von mehreren IT-Projekten ergeben. Hier muss ein **Multiprojektmanagement** Ziel- und Aufgabenüberschneidungen sowie Ressourcenkonflikte verhindern.[345]

---

[341] Lange/Wall/*Picot*, Risikomanagement nach dem KonTraG, § 1 Rdn. 66; Lange/Wall/*Kindler*/*Pahlke*, Risikomanagement nach dem KonTraG, § 1 Rdn. 208, 229.

[342] Lange/Wall/*Kindler*/*Pahlke*, Risikomanagement nach dem KonTraG, § 1 Rdn. 230.

[343] *Seibold*, IT-Risikomanagement, 81; *Koch*, IT-Projektrecht, Rdn. 621.

[344] *Seibold*, IT-Risikomanagement, 18.

[345] *Seibold*, IT-Risikomanagement, 28.

Zu überwachungsbedürftigen Risiken gehören auch Risiken aus dem Ausfall von IT-Systemen[346] durch vorhersehbare Systemfehler, Angriffe aus dem Netz und in Einzelfällen auch Risiken aus der Nutzung nicht lizenzierter Software durch Mitarbeiter. Die Konsequenzen von Störungen und Ausfällen betrieblicher IT-Systeme oder aus massiven Angriffen Dritter aus dem Internet auf das IT-System können zur zeitweisen Unterbrechung der gesamten Produktion und/oder Geschäftstätigkeit führen (insbesondere bei unzureichender Datensicherung) und damit rasch zur Insolvenz des betroffenen Unternehmens. Deshalb gehört auch die Überwachung des gesicherten Ablaufs der Funktionen betrieblicher IT-Systeme zur notwendigen Bestandssicherung, ebenso aber das Vorbeugen von Risiken aus Verzögerung oder gar Scheitern von IT-Projekten.[347]

Teil V

ITIL und ISO-Normen als Prüfmaßstab

ITIL (Information Technology Infrastructure Library“) stellt eine aus der Praxis gebildete „Best Practice“-Sammlung und einen de facto-Standard dar[348], ISO 20000 eine international geltende Norm[349]. Gegenwärtig wird Version ITILv3 angewendet. Die Standards nach ITIL und ISO definieren konkrete Mindestanforderungen, die als objektiver Maßstab für die Ziele von Umsetzungsprojekten dienen können, und liefern

[346] Diese Risiken gelten als „leistungswirtschaftliche“ Risiken (Lange/Wall/*Lange*, Risikomanagement nach dem KonTraG 2001, § 2 Rdn. 22).
[347] *Koch*, IT-Projektrecht, Rdn. 624
[348] *Koch*, IT-Projektrecht, Rdn. 650.
[349] *Koch*, IT-Projektrecht, Rdn. 704.

Kontrollziele zur Überprüfung der Wirksamkeit der Managementprozesse.[350] An diesem Maßstab sind zum einen die Vereinbarungsinhalte von IT-Projektverträgen zu messen, zm anderen die Pflichterfüllung der Geschäftsleitung.

ITIL beschreibt Prozesse, während ISO 20000 Kriterien festlegt, die sich messen lassen, etwa Umfang, Art und Definition der Services, Planung und Implementierung eines neuen oder veränderten Service.[351] Von 216 befragten Unternehmen setzten 80% ITIL ein.[352]

ITIL beschreibt nämlich nur das „Was", nicht das „Wie"[353], gibt also eine Checkliste der erforderlichen Regelungspunkte, legt aber nicht fest, wie die konkreten Regelungen aussehen sollen (was auch angesichts der Unterschiedlichkeit der Sachverhalte nicht viel Sinn machen würde). Sache des Auftraggebers ist, für die jeweiligen Regelungsthemen mit dem Anbieter Vereinbarungen auszugestalten, die den spezifischen Unternehmensanforderungen entsprechen. Dies bedeutet in der **Vertragspraxis**: Wenn die ITIL-Anforderungen für das Change Management etwa ein Überwachen des Umsetzungserfolgs verlangen, so ist hiermit noch kein spezifischer Prüfprozess definiert. Diese Definition und die konkrete Ausgestaltung der IT-Prozesse müssen die Vertragspartei vereinbaren[354]. Die ITIL-Vorgaben können also nicht fehlende Vereinbarungsinhalte ersetzen, sondern nur helfen, solche Versäumnisse zu vermeiden, und zugleich sicherstellen, dass in allen IT-Projektverträgen die jeweils erforderlichen Regelungen verhandelt werden.

Weicht die Geschäftsleitung von den durch ITIL oder ISO-Normen festgelegten Mindestanforderungen ab, trifft sie also z.B. nicht bestimmte vertragliche Regelungen im Projektvertrag oder kontrolliert sie die Erfüllung des vertraglichen Pflichtenprogramms des Anbieters nicht (ausreichend), trägt sie die Beweislast dafür, dass eine nachteilige Entwicklung unabhängig von dieser Abweichung eingetreten wäre.

Für den Leistungsbereich Mängelhaftung und Wartung/Pflege relevant sind insbesondere das Incident Management und das Problem Management (s. unten...).

Service Level Management[355]

Am Beispiel des Service Level Management (SLM) sei die ITIL-Konzeption verdeutlicht. Das SLM soll eine laufende Kontrolle der (kontraktierten) Service

---

[350] BSI, ITIL und Standards für IT-Prozesse, 9 16.

[351] S. den Überblick bei *Steinberg*/*Goodwin*, Computerwoche 2/2007, 24, 25.

[352] *Gammel*, ITIL bringt IR-Abteilungen auf Trab, Computerwoche 17/2005, 34. *Zielke*/*Überhorst*, Computerwoche 27/2005, 14; Bundesamt für Sicherheit in der Informationstechnik (BSI), ITIL und Informationssicherheit – Möglichkeiten und Chancen des Zusammenwirkens von IT-Sicherheit und IT-Service-Management 2005, www.bsi.bund.de, 13.

[353] *Kopperger/Kunsmann/Weisbecker*, IT-Servicemanagement, in: Tiemeyer (Hrg.), Handbuch IT-Management, 115, 134.

[354] *Kopperger/Kunsmann/Weisbecker*, IT-Servicemanagement, in: Tiemeyer (Hrg.), Handbuch IT-Management, 139.

[355] Ausführlich *Koch*, IT-Projektrecht, Rdn. 672.

Levels[356] (SLs) ermöglichen. Mit Kunden schließt der Anbieter Service Level Agreements (SLAs), mit seinen Lieferanten Underpinning Contracts (UCs). Beide Vertragspartner müssen die Service Levels gemeinsam definieren und realisieren; Leistungsmessungen und Transparenz sind von wesentliche Bedeutung.[357] Wesentliche Leistungsparameter sind etwa Verfügbarkeit, Ausfallzeit, maximal tolerierbarer Datenverlust.[358]

Anforderungen nach ITIL:[359]

- Service Level-Anforderung seitens Business und Kunden verhandeln und vereinbaren
- Monitoring und Berichten über tatsächlich erbrachte Service Level-Leistungen (SL-Reporting)
- Laufende Verbesserung der Service Level planen und durchführen (Kontinuierlicher Vervesserungsprozess, KVP).
- Service Level Management und –Support Funktionen koordinieren
- Durchführen von Service Review-Meetings mit Kunden
- Durchführen des Service-Verbesserungsprogramms
- Überwachen der sich ändernden Anforderungen des Unternehmens und entsprechende Anpassung des Service Level Agreements
- Anpassung des Operation Level Agreements (OLAs) und der Unterstützungsverträge UCs) mit externen Lieferanten
- Erstellen und Pflege des Service-Katalogs

Messkriterien

- Abdeckungsgrad des IT-Services
- Anzahl der Abweichungen zu den vereinbarten Service Levels
- Kundenzufriedenheit
- (IT-)Prozesskosten

ISO 20000

Die Konzeption der Norm ISO 20000 soll am Beispiel des Change Management erläutert.[360] Ziel ist, dass alle Changes in kontrollfähiger Weise geprüft, gebilligt, implementiert und überprüft werden. Alle Changes sind nach Implementation auf Erfolg

---

[356] In ITIL werden Service Level Agreements definiert als: „A written agreement or contact between the customers and the IT provider which documents the agreed service levels for an IT service. Typically it will cover: service hours, service availability, customer support levels, throughputs and terminal response time, restrictions, functionality and the service levels to be provided in a contingency. It may also include security and accounting policy." (zit. nach *Victor*/*Günther*, Optimiertes IT-Management mit ITIL, 107).

[357] *Elsässer*, ITIL einführen und umsetzen, 147.

[358] *Bock*/*Macek*/*Oberndorfer*/*Pumsenberger*, ITIL – Zertifizierung nach BS 15000/ISO 20000, 77.

[359] www .itil.org/itil_071.html

[360] Ausführlich *Koch*, IT-Projektrecht, Rdn. 728.

oder Fehlschlagen zu überprüfen (post-implementation review, ISO/IEC 20000-2 Nr. 9.2.2).

Alle Changes sind (nach ISO/IEC 20000-1 Nr. 9.2, 3. Abs.) aufzuzeichnen und zu klassifizieren, etwa als dringend (urgent), Notfall (emergency), major (wesentlich), minor (geringfügig). Sie sind abzustimmen, zu überprüfen und in kontrollierter Weise zu implementieren (Nr. 9.2, 5. Abs.).

Literatur

Schriften des BSI

Bundesamt für Sicherheit in der Informationstechnik (BSI), ITIL und Informationssicherheit – ITIL und Informationssicherheit – Aspekte der Integration von Incident und Security Management, Version 1.0.1, 2006, Version 1.0.1 Sept. 2006, www.bsi.bund.de, zit. als *BSI, ITIL und Standards für IT-Prozesse*.

Bundesamt für Sicherheit in der Informationstechnik (BSI), ITIL und Standards für IT-Prozesse – Prozess-Standards für die Entwicklung der IT-Service-Organisation gemäß ITIL Best Practices, Version 1.0.1, herausgegeben von der Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung, KBSt-Brief 1/2006, Okt. 2006, www.bsi.bund.de, zit. als *BSI, ITIL und Informationssicherheit*

Weitere Literatur

Balzert, Lehrbuch der Software-Technik. Software-Entwicklung, 2. Aufl. 2001
Bettinger/*Scheffelt*, Application Service Providing: Vertragsgestaltung und Konfliktmanagement, CR 2001, 729
Blöse/*Pechardscheck*, Die rechtliche Absicherung von IT-Outsourcing-Projekten, CR 2002, 785, 787
Blume, Projektkompass SAP, Arbeitsorientierte Planungshilfen für die erfolgreiche Einführung von SAP-Software, 2. Aufl. 1998
Bock/*Macek*/*Oberndorfer*/*Pumsenberger*, ITIL – Zertifizierung nach BS 15000/ISO 20000, 2006
Börner/*Buhl*/*Hellmich*/*Klett*/*Moos*, Leitdaten IT-Recht. Ein Handbuch für die Unternehmenspraxis2004
Braun, Die Zulässigkeit von Service Level Agreements – am Beispiel der Verfügbarkeitsklausel, 2006
Brössler/*Siedersleben* (Hrg.), Softwaretechnik, 1.Aufl. 2000
Buchta/*Eul*/*Schulte-Croonenberg*, Strategisches IT-Management, Wert steigern, Leistung steuern, Kosten senken, 2. Aufl. 2005
Dietrich/*Schirra*, IT im Unternehmen, Leistungssteigerung bei sinkenden Budgets – Erfolgsbeispiele aus der Praxis 2004
Ellenberger/Müller, Zweckmäßige Gestaltung von Hardware-, Software- und Projektverträgen, 2. Aufl. 1984
Elsässer, ITIL einführen und umsetzen, Leitfaden für effizientes IT-Management durch Prozessorientierung, 2. Aufl. 2006
Hansen/*Neumann*. Wirtschaftsinformatik 1, Grundlagen und Anwendungen, 9.Aufl. 2005
Heilmann/*Etzel*/*Richter*, IT-Projektmanagement. Fallstricke und Erfolgsfaktoren, Erfahrungsberichte aus der Praxis, 2. Aufl. 2003
Heussen, Richtige Vertragsgestaltung und Ablaufkontrolle bei EDV-Projekten, RWS-Skript 259, 1993
Heussen/*Hoh*, Controlling von EDV-Projekten. Vertragsgestaltung und Kostenkontrolle, RWS-Skript 233, 1992
Hindel/*Hörmann*/*Müller*/*Schmied*, Basiswissen Software-Projektmanagement, 2.Aufl. 2006
Hörl/*Häuser*, Service Level Agreements in IT-Outsourcingverträgen, CR 2003, 713
Intveen/*Lohmann*, Das IT-Pflichtenheft, CR 2003, 640
Klotz/*Dorn*, Vertragsmanagement in der Informationsverarbeitung, Handbuch für Planung, Durchführung und Controlling von IT-Verträgen 2006
Koch, Handbuch Software- und Datenbank-Recht 2003
Koch, Computer-Vertragsrecht, 6. Aufl. 2002
Koreimann, Grundlagen der Software-Entwicklung, 3.Aufl. 2000
Liggesmeyer, Software-Qualität. Testen, Analysieren und Verifizieren von Software2002
Müller-Hengstenberg, BVB-Computersoftware, Besondere Vertragsbedingungen für die Überlassung, Erstellung, Planung und Pflege von DV-Programmem, 3. Aufl. 1992
Müller-Hengstenberg, Der Vertrag als Mittel des Risikomanagements, CR 2005, 385
Müller-Hengstenberg/Krcmar, Mitwirkungspflicht des Auftraggebers bei IT-Projekten, CR 2002, 549
Redeker (Hrg.), Handbuch der IT-Verträge, Stand 2006

Röhrborn/*Sinhart*, Application Service Providing – juristische Einordnung und Vertragsgestaltung, CR 2001, 69

Schneider/*v.Westphalen*, Software-Erstellungsverträge 2006

Schoengarth, Application Service Providing. Vertragsgestaltung und Risiken, insbesondere Betriebsausfallschäden 2005

Schreibauer/*Taraschka*, Service Level Agreements für Softwarepflegeverträge, CR 2003, 557

Schricker/*Loewenheim*, Urheberrecht, 3.Auflage 2006

Seibold, IT-Risikomanagement 2006

Söbbing, Handbuch IT-Outsourcing. Rechtliche, strategische und steuerliche Fragen 2002

Stahlknecht/*Hasenkamp*, Einführung in die Wirtschaftsinformatik, 11. Aufl. 2005

Streitz, IT-Projekte retten. Risiken beherrschen und Schieflagen beseitigen 2004

Thewalt, Der Softwareerstellungsvertrag nach der Schuldrechtsreform. Rechtsnatur, Leistungsbestimmung und Mängelhaftung 2004

Tiemeyer (Hrg.), Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis 2006

Wenzel, Betriebswirtschaftliche Anwendungen des integrierten Systems SAP R/3, 1996

Gräfin v. Westerholt/*Berger*, Der Application Service Provider und das neue Schuldrecht, CR 2002, 81

Zahrnt, Projektmanagement von IT-Verträgen 2002